

TRABALLO FIN DE GRAO
GRAO EN ENXEÑARÍA INFORMÁTICA
MENCIÓN EN SISTEMAS DE INFORMACIÓN

Aplicación multiplataforma para a xestión de venda, adopción e perda de animais domésticos

Estudante: Emilio Paz Sotelo
Dirección: Miguel Rodríguez Penabad

A Coruña, setembro de 2019.

A meus pais

Agradecementos

En primeiro lugar agradecerlle a Miguel Penabad os seus consellos para sacar adiante este traballo de fin de grao. Agradecerlle a meus pais toda a axuda e educación que me deron, sen eles non houbera chegado aquí nunca e agradecerlle a miña nai todos eses ánimos cando non tiña forzas para continuar. Lévome grandes amigos desta facultade, que fixeron que as tardes aquí fora como estar na casa. Especialmente agradecerlle a Christopher a súa axuda en todo, por estar sempre ahí. Moitas grazas a todos de corazón.

Resumo

O obxectivo principal de este proxecto é desenvolver unha aplicación que nos permita poder adoptar, poñer en adopción, denunciar a desaparición dun animal ou crear unha alerta para posibles adopcións dependendo do tipo de animal e/ou raza que desexemos.

A aplicación tamén conterá comentarios en cada anuncio que permitirán comunicarnos cos usuarios e informar sobre o anuncio en cuestión.

Por último, incluírá un pequeno foro no que os usuarios poderán manter charlas sobre diferentes temas acerca dos animais como cruces, vacinas...

O proxecto está composto dunha base de datos na que persistiremos os datos que necesitamos, un servidor que procesará as peticións, unha aplicación web que nos permitirá facer todas as operacións necesarias para tramitar os anuncios e tamén unha aplicación Android na que poderemos consultar os anuncios rapidamente, e así estar atento as últimas novas sobre as perdas ou achegas.

Durante o desenvolvemento do proxecto usaranse tecnoloxías modernas que nos permitan unha interacción rápida, fluída e ao mesmo tempo sinxela para o usuario.

Usouse un ciclo de vida iterativo e incremental. Así, en cada iteración desenvolveuse un modelo en cascada que conta coas etapas de análise, deseño, implementación e probas.

Abstract

The main objective of this project is to develop an application that allows us to adopt, put into adoption, report the disappearance of an animal or create an alert for possible adoptions, depending on the type of animal or breed we want.

The application will have comments on each ad that allow us to communicate with users and inform about these ads.

Finally, it will include a small forum where users can hold talks on different topics about animals, such as crosses, vaccines...

The project consists of a database where we will persist the data we need, a server that will process the requests, a web application that will allow us to perform all the operations necessary to process the ads and also an Android application where we can quickly consult the ads, and so be aware of the latest news about losses or contributions.

During the development of the project, modern technologies will be used that will allow us a fast, fluid and at the same time easy interaction for the user.

An iterative and incremental life cycle was used. Therefore, in each iteration, a cascade model has been developed that has the stages of analysis, design, implementation and testing.

Palabras chave:

- Aplicación móvil
- Android
- React
- Spring-boot
- Spring-data
- Hibernate
- Animales
- Adopción
- Java

Keywords:

- Mobile application
- Android
- React
- Spring-boot
- Spring-data
- Hibernate
- Animals
- Adoption
- Java

Índice Xeral

1	Introdución	1
1.1	Motivación	1
1.2	Obxectivos	1
1.3	Estrutura da memoria	2
2	Estado da arte	3
2.1	Chuby - adopta un perro, gato y más mascotas	3
2.2	Wizapet	3
2.3	Adapan	3
2.4	Mundo Animalia	6
2.5	Conclusións	6
3	Ferramentas e tecnoloxías utilizadas	7
3.1	Para o servizo web	7
3.2	Para a aplicación web	7
3.3	Para a aplicación móbil	8
3.4	Outras ferramentas	8
4	Metodoloxía de desenvolvemento	11
4.1	O proceso unificado de desenvolvemento de software	11
4.1.1	Características	11
4.1.2	Análise Inicial	12
4.1.3	Fases dunha iteración	12
4.1.4	Cerre do proxecto	13
5	Planificación e seguimento	15
5.1	Tarefas e recursos	15
5.1.1	Recursos Humanos	16

5.1.2	Recursos Materiais	16
5.2	Planificación inicial e coste	17
5.3	Seguimento	17
6	Iteración 1	21
6.1	Análise	21
6.1.1	Perfís de usuario	21
6.1.2	Modelo conceptual de datos	22
6.2	Deseño	22
6.2.1	Arquitectura global do sistema	24
6.2.2	Patróns de deseño	24
6.2.3	Modelo lóxico de datos	28
6.3	Implementación	31
6.3.1	Servidor web - Spring Data JPA	31
6.3.2	Aplicación web - React	32
6.3.3	Aplicación móbil - Android	32
6.4	Probas	32
6.4.1	Servidor web	32
6.4.2	Aplicación web - React	33
6.4.3	Aplicación móbil - Android	33
7	Iteración 2	35
7.1	Análise	35
7.1.1	Casos de uso - Xestión de usuarios, razas e tipos de animais	36
7.2	Deseño	41
7.2.1	Deseño da aplicación - Servidor Web	41
7.2.2	Deseño da aplicación - Servizo Web	42
7.3	Implementación	43
7.3.1	Servidor web	43
7.3.2	Aplicación web - React	43
7.4	Probas	43
7.4.1	Servidor web	43
7.4.2	Aplicación web - React	43
8	Iteración 3	51
8.1	Análise	51
8.1.1	Casos de uso - Xestión de anuncios	51
8.2	Deseño	56

8.2.1	Deseño da aplicación - Servidor web	56
8.2.2	Deseño da aplicación - Servizo Web	56
8.2.3	Deseño da aplicación - Cliente Móbil	56
8.3	Implementación	62
8.3.1	Servidor web	62
8.3.2	Aplicacion web - React	62
8.3.3	Aplicacion móbil - Android	62
8.4	Probas	62
8.4.1	Servidor web	62
8.4.2	Aplicacion web - React	62
8.4.3	Aplicacion móbil - Android	62
9	Iteración 4	63
9.1	Análise	63
9.1.1	Casos de uso - Xestión do foro	63
9.2	Deseño	66
9.2.1	Deseño da aplicación - Servizo Web	66
9.2.2	Deseño da aplicación - Cliente Móbil	68
9.3	Implementación	68
9.3.1	Servidor web	68
9.3.2	Aplicación web - React	68
9.3.3	Aplicación móbil - Android	71
9.4	Probas	71
9.4.1	Servidor web	71
9.4.2	Aplicación web - React	71
9.4.3	Aplicación móbil - Android	71
10	Conclusións e traballo futuro	73
10.1	Conclusións	73
10.2	Traballo futuro	74
	Aplicación web	77
	Aplicacion móbil	81
	Relación de Acrónimos	83
	Bibliografía	85

Índice de Figuras

2.1	Lista de anuncios na aplicacion Chuby	4
2.2	Páxina de filtros na aplicación Chuby	4
2.3	Lista de filtros na aplicacion Wizapet	5
2.4	Mapa de desaparición de animais	5
2.5	Lista de animais perdidos	5
2.6	Lista de animais dispostos para a adopción	5
2.7	Lista de anuncios en Mundo Animalia	6
2.8	Exemplo do foro de Mundo Animalia	6
5.1	Diagrama de Gantt	18
6.1	Actores do sistema	22
6.2	Modelo Entidade Relación Estendido	23
6.3	Diagrama de compoñentes da aplicación	25
6.4	Diagrama de clases completo da aplicación	26
6.5	Diagrama UML do Patrón Adaptador	28
7.1	Casos de uso da iteración 2	36
7.2	Diagrama de clases da iteración 2	44
7.3	Entidades do dominio da iteración 2. Interface repositorios	45
7.4	Diagrama dos servizos da iteración 2	46
7.5	Diagrama dos compoñentes usados no servizo web na iteración 2	46
7.6	Maqueta do formulario de rexistro na aplicación web	47
7.7	Maqueta do formulario de login na aplicación web	48
7.8	Maqueta do formulario da xestión de tipos de animais da aplicación web	49
7.9	Maqueta do formulario da xestión de razas da aplicación web	50
8.1	Casos de uso da iteración 3	52

8.2	Diagrama de clases da iteración 3	57
8.3	Diagrama dos servizos da iteración 3	58
8.4	Diagrama dos compoñentes usados no servizo web na iteración 3	59
8.5	Maqueta do listado de anuncios da aplicación web	59
8.6	Maqueta dun anuncio particular da aplicación web	60
8.7	Maqueta da lista de anuncios filtrados por un tipo en particular	61
9.1	Casos de uso da iteración 4	63
9.2	Diagrama dos servizos da iteración 4	67
9.3	Diagrama dos compoñentes usados no servizo web na iteración 4	68
9.4	Maqueta dos diferentes fíos do foro que temos na aplicación	69
9.5	Maqueta dunha interface que representa a vista dun fío do foro na aplicación .	70
1	Páxina de rexistro da aplicación web	77
2	Páxina de login da aplicación web	78
3	Páxina de inicio da aplicación web	78
4	Páxina de rexistro dun animal na aplicación web	79
5	Páxina de rexistro dun anuncio na aplicación web	79
6	Páxina de animais encontrados da aplicación web	80
7	Páxina dun anuncio en particular da aplicación web	80
8	Páxina dun fío en particular da aplicación web	81
9	Páxina de inicio da aplicación aplicación móbil	81
10	Páxina primeria dun anuncio en particular da aplicación móbil	82
11	Páxina segunda dun anuncio en particular da aplicación móbil	82
12	Páxina da lista de fío da aplicación móbil	82
13	Páxina dun fío en particular da aplicación móbil	82

Índice de Táboas

5.1	Recursos humanos e taxas	16
5.2	Gastos totais hardware e software	17
5.3	Custo recursos humanos	17
5.4	Seguimento	19
6.1	Usuario	28
6.2	Anuncio	29
6.3	Comentario anuncio	29
6.4	Anuncio	29
6.5	Animal	29
6.6	Foto	30
6.7	Fío	30
6.8	Comentario foro	30
6.9	Busca adopción	30
6.10	Busca por raza	31
6.11	Busca por tipo animal	31
6.12	Tipo animal	31
6.13	Raza	31

Introdución

1.1 Motivación

Na actualidade, hai máis de 20 millóns de mascotas en España, cifra en aumento ano tras ano, o que nos leva a pensar que son unha das partes máis importantes da nosa vida. Cada día vemos en medios de comunicación, xa sexa na televisión ou en redes sociais, novas sobre desaparicións de animais ou a súa posta en adopción. Isto fai que dita información este moi dispersa por ferramentas cuxo uso principal non é a comunicación deste tipo de novas. Xurde entón a problemática de a qué medio acceder en caso de que se queira obter ou publicar este tipo de sucesos.

Debido ao impacto dos animais na nosa vida, xorden multitude de dúbidas sobre os seus cuidados e non se ten un lugar de referencia onde obter esta información.

E por isto que xurde a idea de desenvolver un lugar de referencia onde poder xestionar todo o relativo as nosas mascotas, xa sexa a súa adopción, desaparición ou información sobre o seu coidado.

1.2 Obxectivos

O obxectivo principal deste proxecto é deseñar e implementar unha ferramenta multiplataforma que este composta por:

- Aplicación web onde os usuarios poidan realizar a xestión integral de todo o relativo as súas mascotas, xa sexa para a súa adopción ou como fonte de información. Os usuarios poderán rexistrarse para publicar anuncios.

A súa vez, esta aplicación tamén conterá un foro onde os usuarios rexistrados poderán obter resposta as súas dúbidas.

- Aplicación móbil que permitirá aos usuarios acceder aos anuncios e ao foro da plataforma sen necesidade de estar diante dun ordenador.
- Implantación dun servidor web que permita o acceso das anteriores aplicacións usando a arquitectura REST.

1.3 Estrutura da memoria

A memoria componse dos seguintes apartados:

- **Estado da arte:** Neste apartado explicaremos brevemente as ferramentas que existen na actualidade para resolver estes problemas.
- **Ferramentas e tecnoloxías utilizadas:** Ferramentas, tecnoloxías e plugins usados no desenvolvemento do proxecto.
- **Metodoloxía de desenvolvemento:** Neste capítulo indicárase a metodoloxía que aplicamos para levar a cabo o proxecto. Veranse as fases e as etapas que forman parte desta metodoloxía.
- **Planificación e seguimento:** Planificación inicial, estimación de custos, diagrama de Gantt e seguimento do proxecto.
- **Iteración 1:** Neste capítulo, e nas seguintes iteracións, seguiremos as etapas de análise, deseño, implementación e probas do Proceso Unificado de Desenvolvemento Software. Nesta iteración farase un estudo dos actores, desenvolverase o modelo conceptual do proxecto e describiremos a distribución dos datos que vamos almacenar no noso sistema.
- **Iteración 2:** Nesta iteración expoñerase a xestión de usuarios na aplicación, podendo boralos, crealos ou editalos. Tamén se xestionarán as razas e os tipos de animais aos que pertencen.
- **Iteración 3:** Nesta iteración expoñerase a creación, actualización e borrado dos anuncios de animais.
- **Iteración 4:** Levouse a implementación do foro e a xestión dos seus fíos e comentarios por parte dos usuarios.
- **Conclusións e traballo futuro:** Por último expóñense as valoracións finais sobre a aplicación realizada e tamén posibles melloras para un futuro.

Estado da arte

Analizaranse catro ferramentas que existen no mercado pero que presentan algunhas carencias con respecto ao que queremos para este proxecto. Estudaremos 4 aplicacións, na que dúas serán web e outras dúas Android.

2.1 Chuby - adopta un perro, gato y más mascotas

Aplicación Android que nos permite crear anuncios sobre desaparición e achegas de animais e tamén a posta en adopción destes. Non nos permite crear anuncios para poder solicitar a adopción dalgún animal. Tampouco posúe un foro que nos permita discutir temas que nos interesen e non está enfocada para localizacións en España. Vemos na figura 2.1 a lista de anuncios con fotografías cada animal e un título indicando de que anuncio se trata. Na figura 2.2 pódense observar os filtros dispoñibles, que nos permiten crebar a lista de anuncios dependendo de diversos criterios.

2.2 Wizapet

Aplicación Android para poder localizar a través dun mapa perdas, adopcións e animais encontrados. Non se poden crear anuncios para querer adoptar un animal nin tampouco posúe un foro para poder comentar novas. Pode resultar un pouco confuso o mapa debido a multitude de opcións que se nos presentan, como vemos nas figuras 2.3 e 2.4,

2.3 Adapan

Aplicación web na que se ofrecen varias opcións para os animais, como ver unha lista de perdas (Figura 2.5) ou unha lista de adopcións, (Figura 2.6). Como carencia, só nos permite a opción dun animal e tampouco ten a posibilidade dun foro.



Figura 2.1: Lista de anuncios na aplicación Chuby

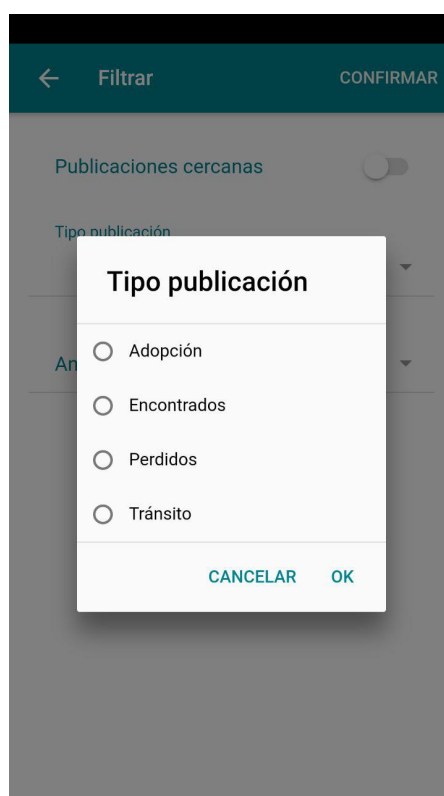


Figura 2.2: Páxina de filtros na aplicación Chuby

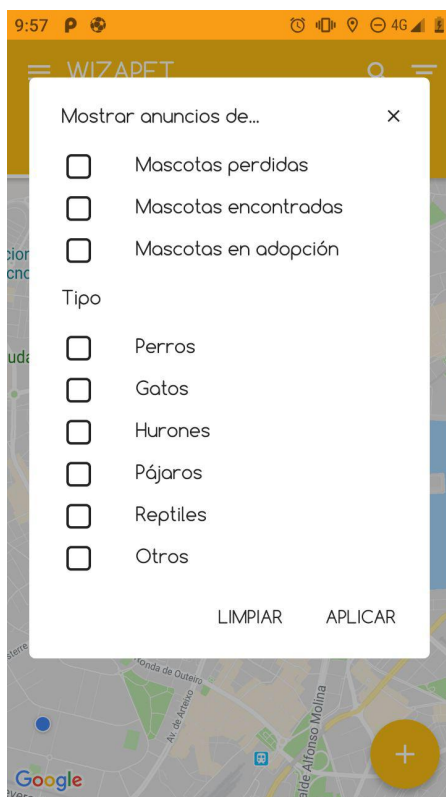


Figura 2.3: Lista de filtros na aplicación Wizapet



Figura 2.4: Mapa de desaparición de animais



Figura 2.5: Lista de animais perdidos

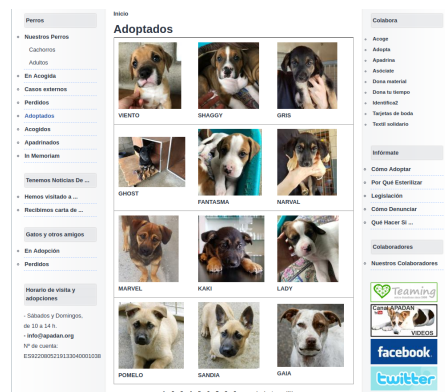


Figura 2.6: Lista de animais dispostos para a adopción

2.4 Mundo Animalia

Aplicación web na que se ofrece a posibilidade de consultar unha lista de posibles adopcións (Figura 2.7). Tamén se ofrece a opción de consultar un foro (Figura 2.8) e realizar consultas sobre os animais, en diferentes seccións. O que non ofrece *Mundo Animalia* é a posibilidade de crear anuncios para denunciar buscas ou perdas.

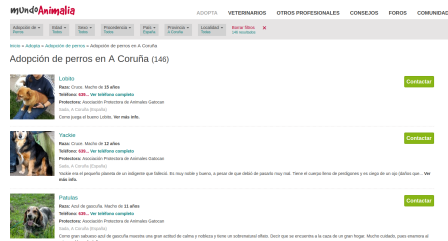


Figura 2.7: Lista de anuncios en Mundo Animalia

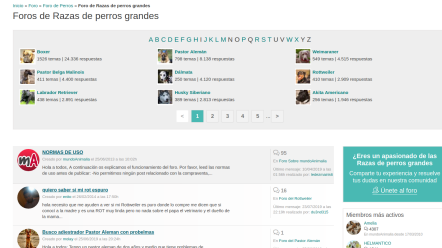


Figura 2.8: Exemplo do foro de Mundo Animalia

2.5 Conclusións

Existe no mercado unha ampla variedade de aplicacións, sobre todo para dispositivos Android. Algunhas ofrecen funcionalidades para crear anuncios de perdas ou de aparicións de animais.

O que non implementan por exemplo estas aplicacións é a posibilidade de crear un anuncio para solicitar a adopción dun mesmo, que nos permita ao mesmo tempo estar alerta de si se produce algunha posibilidade de adopción. Tamén case todas as aplicacións consultadas, sobre todo as Android, necesitan rexistro para poder consultar os anuncios.

En resumo, o principal obxectivo deste proxecto será crear un sistema sinxelo de anuncios de animais e que poidamos consultar os datos rapidamente apoiados por un foro no que comentar novas.

Ferramentas e tecnoloxías utilizadas

Para o desenvolvemento deste proxecto realizouse un servidor web, unha aplicación web e unha móbil nativa. Para explicar as tecnoloxías que usamos, agrupámolas polos diferentes servizos que tivemos que implementar.

3.1 Para o servizo web

- **PostgreSQL:** PostgreSQL[1] é un sistema de xestión de base de datos relacional, orientado a obxectos e de código aberto. Usado no proxecto como motor de base de datos para persistir os datos da aplicación.
- **Java:** Java[2] é unha linguaxe de programación de propósito xeral, concorrente e orientado a obxectos, deseñado principalmente para poder ser executado en calquera dispositivo.
- **Spring:** Spring[3] é un framework para o desenvolvemento de aplicacións e contedor de inversión de control, de código aberto e para a plataforma Java. Por exemplo, mediante anotacións, permitíranos a persistencia dos datos de forma intuitiva e sinxela.
- **JWT:** Estándar aberto baseado en JSON para a creación de *tokens* de acceso. Esta tecnoloxía permítenos transmitir información de forma segura e sinxela entre cliente e servidor. O funcionamento e sinxelo, cando un usuario se identifica correctamente na aplicación, este recibe un *token* que usará nas vindeiras peticións para poder acceder a rutas, servizos e recursos protexidos.

3.2 Para a aplicación web

- **React:** React[4] é unha librería JavaScript de código aberto deseñada para crear interfaces de usuario co obxectivo de facilitar o desenvolvemento dunha aplicación SPA. O

obxectivo principal de React e a creación de compoñentes interactivos e reutilizábeis para esas interfaces de usuario.

- **Material - UI:** Librería JavaScript similar a Bootstrap que nos permite importar compoñentes de React para favorecer un desenvolvemento da web máis sinxelo, interactivo e rápido.
- **HTML:** Código que se utiliza para estruturar e despregar unha páxina web e os seus contidos.
- **CSS:** Código que se utiliza para poder dar estilo a unha páxina web a través das denominadas follas de estilo.
- **JavaScript:** [5] Linguaxe de programación que nos permite realizar actividades complexas nunha páxina web, como contido dinámico, controlar arquivos multimedia, ou crear imaxes animadas.

3.3 Para a aplicación móbil

- **Android:** [6] Sistema operativo desenvolvido por Google, deseñado para teléfonos móbiles baseados no sistema operativo Linux baseado no kernel de Linux. É de carácter libre e gratuito. Por estes motivos decidiuse desenvolver o cliente móbil usando o seu SDK. Para poder implementar unha aplicación Android recoméndase o IDE Android Studio que nos aporta todas as ferramentas para poder desenvolver unha aplicación Android.
- **Retrofit:** Librería para Android que nos permite realizar peticións a un servidor. Xestiona diferentes tipos de parámetros e parsea automaticamente a resposta a un POJO.

3.4 Outras ferramentas

Ferramentas que se usaron para o deseño de interfaces e para modelar diagramas.

- **Dia:** Aplicación libre de código aberto que nos permite realizar gráficos de calquera tipo de forma sinxela, como por exemplo o entidade-relación.
- **Draw.io:** Aplicación web que nos axudou a modelar o modelo relacional facilmente.
- **Smartsheet:** Aplicación web usada para realizar o diagrama de Gantt para establecer a planificación.

- **IntelliJ IDEA e Android Studio:** Entornos de desenvolvemento integrados para desenvolver programas informáticos e para aplicacións Android respectivamente.
- **ninjamock.com:** Aplicación web que axudou para deseñar as maquetas iniciais de deseño.

Metodoloxía de desenvolvemento

UNHA metodoloxía de desenvolvemento ten como obxectivo guiarnos a través dunhas pautas e tratar de conseguir os obxectivos iniciais dun proxecto, todo isto, a través de diferentes etapas. Usouse a metodoloxía do proceso unificado do software debido a súa capacidade para estar suxeita a cambios e a experiencia do usuario nela.

4.1 O proceso unificado de desenvolvemento de software

O proceso unificado de desenvolvemento de software[7, 8] é unha metodoloxía baseada en compoñentes e en interfaces ben definidas, é a metodoloxía máis usada para o análise, deseño, implementación e probas en sistemas orientados a obxectos. Esta metodoloxía esta centrada en que o desenvolvemento sexa un proceso incremental e iterativo.

A continuación, describiranse as diferentes propiedades que posúe esta metodoloxía.

4.1.1 Características

- Iterativo e incremental. No Proceso Unificado de desenvolvemento de Software hai catro fases fundamentais denominadas inicio, elaboración, construción e transición. Estas fases están reflectidas en cada iteración. Cada realización dunha iteración proporciona un incremento das funcionalidades do sistema. A súa vez estas iteracións están divididas nas etapas dun ciclo de vida clásico: Análise, Deseño, Implementación e probas.
- Baseado en casos de uso. Un caso de uso é un fragmento dunha funcionalidade do sistema, na que se proporciona un resultado a un usuario. Estes son utilizados para obter os requisitos funcionais e para definir os contidos das iteracións. Cada iteración componse dun conxunto de casos de uso.
- Centrado na arquitectura. A arquitectura e a representación dunha vista nun deseño, facendo fincapé nas características mais importantes. A arquitectura define o deseño

do sistema. Está relacionado cos casos de uso e permite a representación destes.

- **Enfocado nos riscos.** É de vital importancia que o desenvolvemento se centre nos riscos que poidan xurdir nas primeiras etapas de desenvolvemento para poder afrontalos con marxe.

4.1.2 **Análise Inicial**

Nesta fase inicial describírase o alcance do proxecto, que consiste na descrición do produto final, incluíndo os seus requisitos ou características. Tamén se realizará unha planificación do proxecto e unha estimación do custo do mesmo. Por último, farase un estudo das tecnoloxías a usar na realización. Esta etapa será constante ao longo do proxecto, xa que durante o desenvolvemento do mesmo á aprendizaxe será algo continuo.

4.1.3 **Fases dunha iteración**

Este proxecto desenvólvese seguindo un proceso iterativo, exactamente en 4 iteracións, que se indican nos capítulos 6, 7, 8 e 9 respectivamente. Cada iteración estará composta das fases que se explican a continuación:

- **Análise** Nesta primeira fase analizaranse as necesidades dos usuarios finais para poder determinar os obxectivos que deber cumprir o proxecto. Establécese unha especificación completa dos requisitos que determinará o que debe facer o sistema sen entrar en particularidades.
- **Deseño** Na fase de deseño, unha vez establecidos os requisitos na fase anterior, obtérase unha solución para poder levalos a cabo. Nesta etapa, deseñaranse os diagramas de clases ou secuencia e o modelo relacional da base de datos, e intentarase deixar claro o funcionamento interno da aplicación como tamén os datos que se manexan. O deseño da aplicación terá en conta a separación lóxica dos diferentes módulos se os houbera.
- **Implementación** A continuación, a fase de implementación consiste na codificación do definido na fase anterior. O obxectivo deste apartado é que a súa finalización obteñamos un incremento da aplicación en sí e non que se abordarán os casos de uso que tiñamos fixados para esta iteración.
- **Probas** O obxectivo desta etapa é probar todo o posible o proxecto para atopar o maior número de erros e solucionalos. Realízanse probas unitarias, que consisten en probar cada un dos módulos que compoñen a aplicación por separado.

4.1.4 Cerre do proxecto

Por último realízase un documento de despregue, que consistirá en explicar os pasos para poder implantar e instalar a aplicación nun entorno de produción. Faise tamén unha revisión de toda a memoria que consiste corrixir posibles erros que teñamos. Farase tamén un manual de uso da aplicación e por derradeiro realizarase tamén a presentación necesaria para explicar o produto.

Planificación e seguimento

5.1 Tarefas e recursos

Como se indica no capítulo 4, cada iteración será un módulo independente e que se corresponderá cunha tarefa a efectos de planificación.

As iteracións que se estableceron foron as seguintes:

- **Iteración 1: Definición da arquitectura do sistema** Nesta primeira iteración levarase a cabo un proceso previo de análise da aplicación, onde se indicarán as funcionalidades e os actores que formarán parte do sistema e tamén a arquitectura do sistema, na que se indicará patróns de deseño usados e un entidade relación.
- **Iteración 2: Xestión de usuarios e tipos de animais** Nela levarase a cabo o sistema de rexistro de usuarios e login da aplicación, tamén creación de razas e tipos de animais.
- **Iteración 3: Xestión de anuncios** Creación, actualización, borrado de anuncios de animais por parte dun usuario, así como poder ver o listaxe de todos eles creados por outros membros.
- **Iteración 4: Xestión do foro** Creación do foro, que estará composto por fíos de diversa temática onde a xente rexistrada poderá escribir comentarios.

Cada iteración dividirase en catro apartados, análise, deseño, implementación e probas. Indicar que fora das etapas clásicas da metodoloxía engadimos unha etapa básica, denominada análise inicial, onde se falará do alcance do proxecto, a planificación inicial e un estudo das tecnoloxías.

A duración prevista do proxecto será de aproximadamente 4 meses, dando comezo o día 01/06/2019 e rematando o 01/09/2019 cun total de 473 horas traballadas. A xornada laboral componse de 5 horas cada día.

Con respecto aos recursos utilizados para o desenvolvemento do proxecto, identificaremos dous tipos:

5.1.1 Recursos Humanos

As tarefas realizaranse por unha mesma persoa pero con distintos roles dependendo da situación a afrontar, temos os seguintes:

- **Analista** Encargarase das tarefas de análise e deseño da aplicación
- **Programador** O seu labor será implementar as funcionalidades que contempla a nosa aplicación. Tamén se encargará das probas.

A continuación, vemos un custo aproximado de cada un deles por hora:

Nome do recurso	Capacidade máxima	Taxa estándar
Analista	100%	25€/hora
Programador	100%	15€/hora

Táboa 5.1: Recursos humanos e taxas

5.1.2 Recursos Materiais

Por outro lado, no caso dos recursos materiais, necesitaremos un dispositivo móbil con Android para poder probar a nosa aplicación, este terá un custo de 150€.

Tamén necesitaremos un equipo informático, este ten un custo de 700€, un sistema operativo Ubuntu 16.04, un IDE para implementar o servidor, que será o IntelliJ Idea, usaremos Android Studio para desenvolver a aplicación móbil, e un editor de código fonte para a aplicación web, o Visual Studio Code. Todos os software son libres, a excepción do IntelliJ Idea, na que usaremos a versión que nos proporciona JetBrains para estudantes, sendo o seu custo tamén 0€.

Recurso	Custo
Hardware	850€
Software	0€
Total (sen IVE)	850€

Táboa 5.2: Gastos totais hardware e software

5.2 Planificación inicial e coste

Coñécese como planificación o proporcionar un marco de traballo que nos permite facer estimacións razoables dos recursos e os seus custos. O que significa, que supón un paso clave na elaboración do proxecto.

Na seguinte táboa indícanse os diferentes custos dos recursos humanos previstos para o desenvolvemento deste:

Tarefa	Horas	Rol	Total
Análise	150	Analista	3750€
Deseño	65	Analista	1625€
Implementación	195	Programador	2925€
Probas	55	Programador	825€
Seguimento	8	Analista	160€
Horas totais	473	Custo total (sen IVE)	9.285€

Táboa 5.3: Custo recursos humanos

A través dun diagrama de Gantt, Figura 5.1, expoñeremos o tempo de dedicación previsto para as diferentes tarefas, ao longo dun tempo total determinado.

5.3 Seguimento

Durante o desenvolvemento do proxecto, as tres últimas iteracións sufriron pequenas desviacións Na táboa 5.4 indícase a diferenza entre o custo planificado e o custo real:

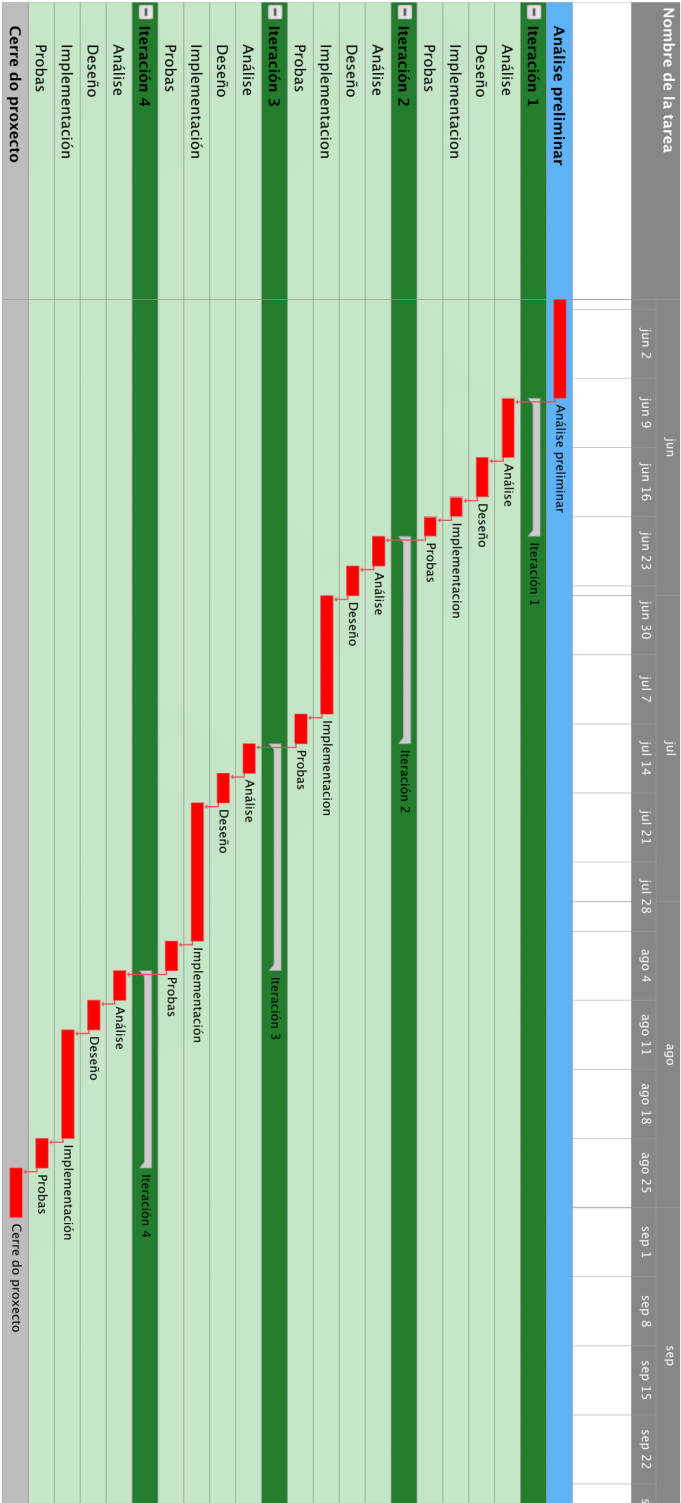


Figura 5.1: Diagrama de Gantt

Iteración	Horas estimadas	Horas realizadas	Desviación
Iteración 1	70	70	0
Iteración 2	105	115	10
Iteración 3	115	125	10
Iteración 4	100	100	0

Táboa 5.4: Seguimento

Como se indica na táboa 5.4 podemos ver que as desviacións foron pequenas debido a un bo proceso de análise da aplicación. As pequenas desviacións das iteracións 2 e 3 foron debido a ser as etapas de máis dificultade na etapa de implementación do proxecto.

Polo tanto, en total sufriuse unha desviación dun 5% con respecto a planificación inicial. Esta desviación supón un aumento do custo final, quedándose nun total de 9585€.

Capítulo 6

Iteración 1

Nesta primeira iteración vaise a realizar un estudo dos actores que teñen incidencia no noso proxecto. Desenvolveremos o modelo conceptual do sistema, no que a través dun diagrama Entidade-Relación Estendido e debido ao seu nivel de abstracción alto, poderemos describir a distribución dos datos que vamos a almacenar no noso sistema.

Debido a metodoloxía usada, o labor para o desenvolvemento do proxecto sepárase en iteracións, como se indicou no capítulo 5.1, pero para ofrecer unha documentación máis sinxela e clara, decidiuse plasmar nesta primeira iteración os diagramas completos, aínda que estes foran realizados por iteracións como estipula a metodoloxía.

No Entidade Relación Estendido indicado neste apartado (Figura 6.2, reflicte a totalidade da base de datos pero, por exemplo, os tipos de entidades usados para a xestión de anuncios, foron realmente incluídos no proxecto na iteración 3, capítulo 8.

6.1 Análise

6.1.1 Perfís de usuario

A continuación mostrarase un estudo dos actores (Figura 6.1), que son aquelas entidades externas que teñen interacción directa co noso sistema. Distinguimos 3 tipos:

- **Usuario sen autenticar:** Un usuario sen identificar correspóndese co perfil de usuario que ten un cliente nada máis entrar na aplicación. Este tipo de perfil poderá consultar os diferentes tipos de anuncios e ver os anuncios asociados a cada un pero, non poderá crealos como tampouco comentar nos mesmos. Tamén terá dispoñible o poder consultar o foro, pero só poderá ver os fíos que teñen visibilidade total.
- **Usuario autenticado:** Trátase do rol que posuirá calquera usuario que estea autenticado na aplicación, grazas a un rexistro previo na mesma. Un usuario rexistrado poderá consultar e crear anuncios, borrarlos e tamén poñer un comentario nestes mesmos.

Tamén terá dereito a crear fíos no foro e a consultalos, e se o desexa tamén, poñer comentarios neles. Por último tamén poderá crear animais para ter un rexistro das súas mascotas.

- **Usuario administrador:** Este tipo de perfil terá as mesmas funcionalidades que un usuario autenticado, e a maiores poderá xestionar as razas e os tipos de animais que temos, podendo crealos ou boralos. Por último, tamén poderá borrar usuarios e a súa información asociada se o desexan.



Figura 6.1: Actores do sistema

6.1.2 Modelo conceptual de datos

Nesta sección indícase o modelo Entidade Relación Estendido (Figura 6.2), que nos permite representar as entidades que van a integrar o noso proxecto na base de datos. Para unha mellor claridade da documentación, decidiuse unificar o Entidade Relación nun só para que nas diferentes iteracións non resulte confuso. Na sección 6.2.3 móstrase un diccionario de datos obtido a partir do modelo relacional, este diccionario permítenos gardar e coñecer o fluxo dos datos dos que fai uso a aplicación.

6.2 Deseño

Nesta sección vanse presentar os diferentes elementos de deseño do sistema, mostrando tres fases destacadas, como son a arquitectura global da aplicación, a lóxica de negocio, e por último, a interface gráfica da aplicación.

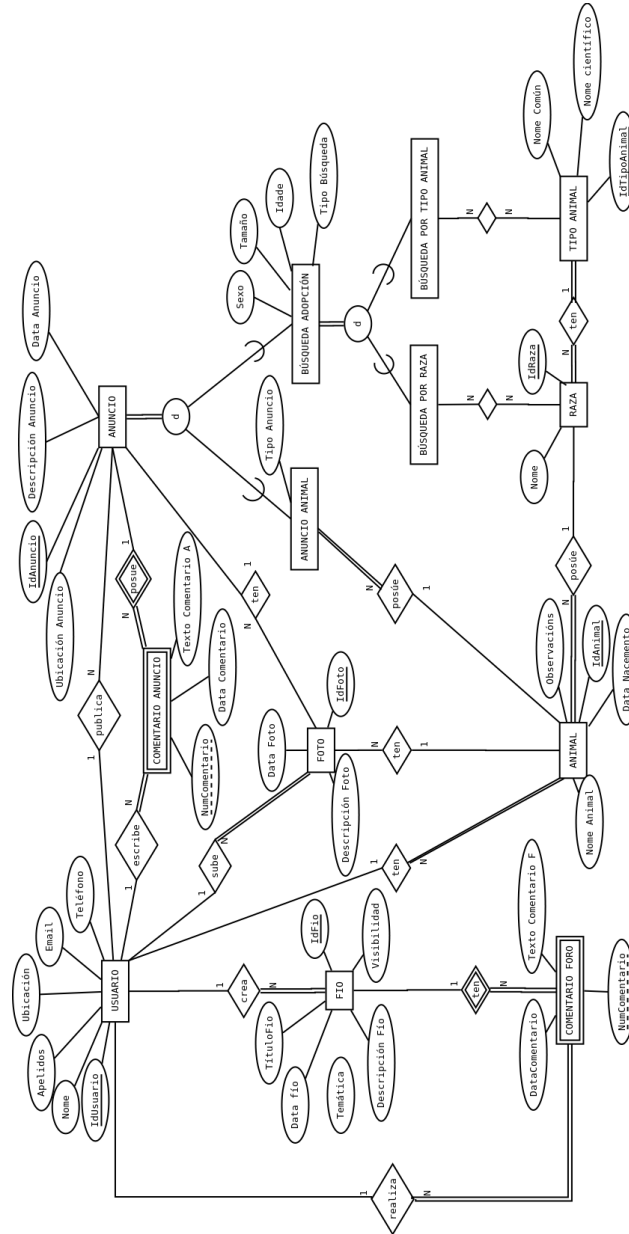


Figura 6.2: Modelo Entidade Relación Estendido

6.2.1 Arquitectura global do sistema

Na Figura 6.3 móstrase un diagrama de compoñentes que intenta representar a división dos compoñentes que temos na aplicación e as súas dependencias entre sí. Para o desenvolvemento deste sistema utilizouse unha arquitectura baseada en tres capas. Unha de datos, outra para o servidor de aplicacións web e a outra para os sistemas de visualización. A arquitectura de tres capas baséase no patrón MVC. Este patrón arquitectónico caracterízase en que separa os datos da lóxica de negocio da súa representación e do módulo encargado de xestionar os eventos e as comunicacións.

A continuación, descríbense estes compoñentes:

- **Cliente web** A través dun navegador poderase consultar e xestionar todas as funcionalidades dispoñibles na aplicación.
- **Cliente Android** A través dun cliente móbil Android poderemos executar a aplicación móbil e xestionar diversas funcionalidades da mesma.
- **Servidor** Parte do sistema que se compón de varios módulos que nos permiten construír xuntos un sistema que permitirá realizar operacións sobre a base de datos. Os diferentes módulos son:
 - **Controlador Rest** Mediante o uso dunha API Rest[9], os clientes poderán consumir as diferentes funcionalidades que ofrece o sistema, este tamén validará as peticións realizadas. Encargarase tamén de invocar as operacións do servizo para obter os datos que necesitemos.
 - **Servizos** Compoñentes onde está encapsulada a lóxica de negocio da aplicación . Está composto de cinco módulos, para a xestión de usuarios, para a de anuncios, para a do foro, para a de animais e para a de envío de emails.
 - **Repositorios** Compoñente da aplicación onde se realizan as operacións de acceso a datos segundo as necesidades.
- **Base de datos** Almacénanse todos os datos relacionados coa xestión da aplicación, como os usuarios, os anuncios, os animais...

6.2.2 Patróns de deseño

Un patrón de deseño[10] é unha técnica para resolver problemas no desenvolvemento de software, axúdanos a resolver problemas de deseño que poidan xurdir. Indícanse a continuación os patróns máis destacados no desenvolvemento deste sistema.

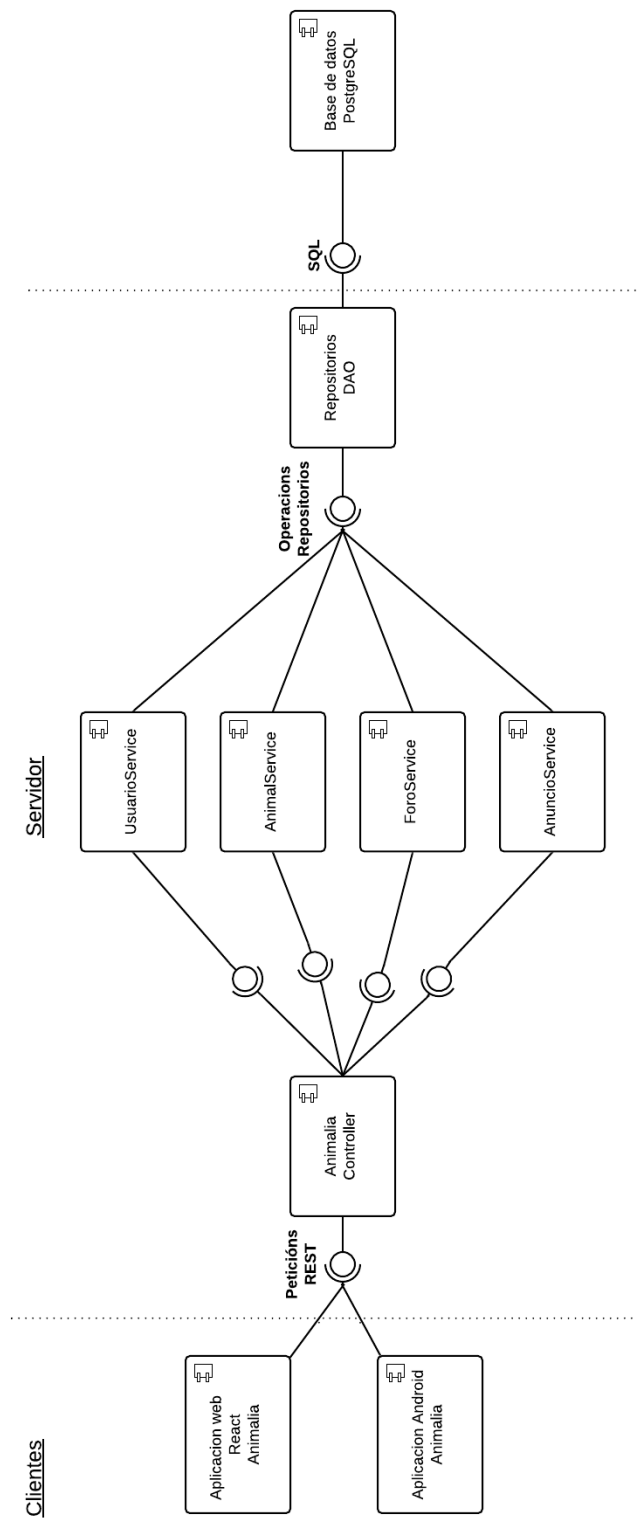


Figura 6.3: Diagrama de componentes de la aplicación



- **Patrón DTO** Un *Data Transfer Object*, DTO, Este patrón permítenos crear un obxecto plano, POJO, con unha serie de atributos que poidan ser enviados ou recuperados dun servidor nunha soa invocación. Isto permítenos que un mesmo obxecto poida conter información de múltiples fontes e centralas nunha única clase simple. Esta clase simple é o que denominamos DTO. Na implementación deste sistema usouse este patrón en varias situacións, un caso foi o de mostrar un anuncio en particular, ese anuncio mostra todos os datos do anuncio en si e tamén mostra os datos do usuario que o creou. Grazas a este patrón podemos enviar estes datos nunha soa petición, a través da clase *AnuncioAnimalDTO*. Na Figura 7.4 podemos ver un exemplo do uso deste patrón.
- **Patrón Repository** Para abstraer estas entidades, úsase o patrón de repositorio [6]. Conceptualmente, un Repositorio encapsula o conxunto de obxectos persistentes nunha base de datos e as operacións realizadas sobre elas, proporcionando unha concepción máis orientada para obxectos de capa de persistencia. Ademais, apoia o obxectivo de lograr unha separación sobre unha dependencia limpa e unidireccional entre o dominio e a capa de acceso a datos. O uso do patrón Repository reduce tamén a cantidade de código necesario para tratar con todas as consultas que se leven a cabo. As clases ou entidades do dominio do proxecto, grazas ao uso do marco Spring Data JPA, fará uso deste patrón estendendo unha interface para proporcionar para operacións CRUD nun repositorio para cada tipo de entidade específica. Na Figura 7.3 vemos un exemplo do uso deste patrón no noso sistema.
- **Patrón Fachada** O uso deste patrón permítenos desacopar o modelo do resto do sistema. A ventaxa de usar este patrón é que nos proveerá dunha interface entre a capa modelo e a capa da lóxica de negocio, permitíndonos unha comunicación máis simplificada entre un módulo e outro. Vemos un uso deste sistema na Figura 8.3.
- **Patrón Adaptador** O patrón adaptador[10] permítenos converter a interface dunha clase en outra que espera realmente o cliente. Este patrón proporcionanos a ventaxa de que estas dúas clases traballen xuntas, xa que doutro modo serían incompatibles. No deseño da parte móbil, deseñouse un patrón adaptador para que a clase *AnuncioAnimal* poidera ser vista en cada fila da lista. Vemos a implementación deste patrón na Figura 6.5.
- **Patrón MVC** O patrón Modelo Vista Controlador (MVC) é un patrón de arquitectura software que separa a lóxica da aplicación (o modelo) da súa representación, (a vista) e para a interacción entre os dous introduce un elemento chamado controlador, que é o responsable de transferir as solicitudes da vista ao modelo para que este faga o que corresponda, e despois o modelo devolva o resultado ao controlador e que sexa este o que envía o resultado á vista para que representase os datos.

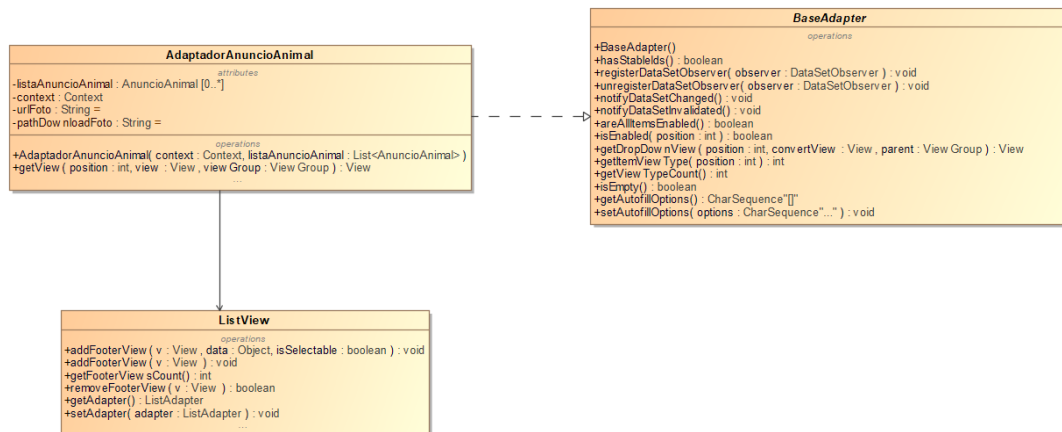


Figura 6.5: Diagrama UML do Patrón Adaptador

6.2.3 Modelo lóxico de datos

Na seguinte Figura 6.4 podemos ver o diagrama de clases completo das entidades do dominio que nos permite describir a estrutura do sistema mostrando as clases do sistema, os seus atributos, operacións e as súas relacións entre si. No apartado 6.2.3 móstrase un diccionario de datos que nos indicará todos os elementos que forman parte do fluxo de datos de todo o sistema. Prescínlese do modelo relacional xa que sería practicamente idéntico ao diagrama de clases exposto.

A continuación móstrase unha explicación dos atributos de cada entidade que teremos no noso sistema. Os atributos subliñados correspóndese coas claves primarias de cada entidade.

Nome do campo	Tipo de dato	Obrigatoriedade	Único
<u>id_usuario</u>	Enteiro	Sí	Sí
nome	Carácter variable	Sí	Non
apelidos	Carácter variable	Non	Non
ubicacion	Carácter variable	Sí	Non
email	Carácter variable	Sí	Sí
telefono	Carácter variable	Sí	Non

Táboa 6.1: Usuario

Nome do campo	Tipo de dato	Obrigatoriedade	Único
<u>id_anuncio</u>	Enteiro	Sí	Sí
ubicacion_anuncio	Carácter variable	Sí	Non
descripcion_anuncio	Carácter variable	Sí	Non
data_anuncio	Data	Non	Non
Claves Foráneas	id_usuario(Usuario)		

Táboa 6.2: Anuncio

Nome do campo	Tipo de dato	Obrigatoriedade	Único
<u>id_comentario_anuncio</u>	Enteiro	Sí	Sí
texto_comentario_anuncio	Carácter variable	Sí	Non
data_comentario_anuncio	Data	Sí	Non
Claves Foráneas	id_usuario(Usuario), id_anuncio(Anuncio)		

Táboa 6.3: Comentario anuncio

Nome do campo	Tipo de dato	Obrigatoriedade	Único
<u>id_anuncio</u>	Enteiro	Sí	Sí
tipo_anuncio	Carácter variable	Sí	Non
Claves Foráneas	id_animal(Animal)		

Táboa 6.4: Anuncio

Nome do campo	Tipo de dato	Obrigatoriedade	Único
<u>id_animal</u>	Enteiro	Sí	Sí
data_nacemento	Data	Non	Non
observacions	Carácter variable	Non	Non
nome_animal	Carácter variable	Non	Non
Claves Foráneas	id_animal(Animal), id_raza(Raza)		

Táboa 6.5: Animal

Nome do campo	Tipo de dato	Obrigatoriedade	Único
<u>id_foto</u>	Enteiro	Sí	Sí
data_foto	Data	Non	Non
descripcion_foto	Carácter variable	Non	Non
datos_foto	Carácter variable	Non	Non
Claves Foráneas	id_usuario(Usuario), id_animal(Animal), id_anuncio(Anuncio)		

Táboa 6.6: Foto

Nome do campo	Tipo de dato	Obrigatoriedade	Único
<u>id_fio</u>	Enteiro	Sí	Sí
titulo_fio	Carácter variable	Sí	Non
descripcion_foto	Carácter variable	Sí	Non
data_fio	Data	Sí	Non
tematica	Carácter variable	Non	Non
Claves Foráneas	id_usuario(Usuario)		

Táboa 6.7: Fío

Nome do campo	Tipo de dato	Obrigatoriedade	Único
<u>id_comentario_foro</u>	Enteiro	Sí	Sí
texto_comentario_foro	Carácter variable	Sí	Non
data_comentario_foro	Data	Sí	Non
Claves Foráneas	id_fio(Fío), id_usuario(Usuario)		

Táboa 6.8: Comentario foro

Nome do campo	Tipo de dato	Obrigatoriedade	Único
<u>id_anuncio</u>	Enteiro	Sí	Sí
idade_busca	Enteiro	Non	Non
sexo	Carácter variable	Non	Non
peso	Enteiro	Non	Non
tipo_busca	Carácter variable	Sí	Non
Claves Foráneas			

Táboa 6.9: Busca adopción

Nome do campo	Tipo de dato	Obrigatoriedade	Único
<u>id_anuncio</u>	Enteiro	Sí	Sí
<u>id_raza</u>	Enteiro	Sí	Sí
Claves Foráneas	id_anuncio(Anuncio)		

Táboa 6.10: Busca por raza

Nome do campo	Tipo de dato	Obrigatoriedade	Único
<u>id_anuncio</u>	Enteiro	Sí	Sí
<u>id_tipo_animal</u>	Enteiro	Sí	Sí
Claves Foráneas	id_anuncio(Anuncio)		

Táboa 6.11: Busca por tipo animal

Nome do campo	Tipo de dato	Obrigatoriedade	Único
<u>id_tipo_animal</u>	Enteiro	Sí	Sí
nome_comun	Carácter variable	Non	Sí
nome_cientifico	Carácter variable	Non	Sí
Claves Foráneas			

Táboa 6.12: Tipo animal

Nome do campo	Tipo de dato	Obrigatoriedade	Único
<u>id_raza</u>	Enteiro	Sí	Sí
nome_raza	Carácter variable	Non	Sí
Claves Foráneas	id_tipo_animal(TipoAnimal)		

Táboa 6.13: Raza

6.3 Implementación

Neste apartado levouse a cabo a configuración do entorno de desenvolvemento necesario para realizar o traballo nas seguintes iteracións. A continuación explicaremos as diferentes tecnoloxías empregadas nesta fase.

6.3.1 Servidor web - Spring Data JPA

Para o desenvolvemento do servidor web, especialmente para a implementación da persistencia, usouse o framework Spring Data JPA[11].

Por cada entidade créase un repositorio, o que nos proporciona unha serie de operacións CRUD que podemos realizar cara a base de datos. Isto permítenos non ter que realizar a implementación dun típico DAO e así axilizar o uso desas operativas. No apartado de patrons de deseño, 6.2.2, coméntase este patrón en máis profundidade.

Para usar estes repositorios, debemos usar anotacións nas entidades que queremos que sexan persistentes na base de datos, e marcar con estas notacións, as relacións que sexan necesarias entre estas entidades. Isto permitirá que se creen as táboas no esquema de base de datos. Así, cando declaremos os repositorios por cada entidade, só teremos que declarar na cabeceira que estamos facendo referencia a esa entidade e o id polo que se insertará na base de datos. Na Figura 7.3 vemos un diagrama que mostra o uso de Repositories no noso sistema. Un exemplo dunha implementación dun repositorio podemos velo no seguinte código fonte.

```

1      @Repository
2      public interface AnuncioAnimalRepository extends
      JpaRepository<AnuncioAnimal,Long> {
3          List<AnuncioAnimal> findByTipoAnuncio(TipoAnuncio
      tipoAnuncio);
4      }
5

```

6.3.2 Aplicación web - React

Para o desenvolvemento da aplicación web, usouse o sistema de xestión de paquetes npm, o que nos permite crear mediante o comando *npm create-react-app* unha web SPA básica, sen ter que realizar ningún tipo de configuración a maiores.

6.3.3 Aplicación móbil - Android

Por último, para o desenvolvemento da aplicación móbil, usouse o IDE Android Studio, que grazas a que xa é un software destinado para o desenvolvemento en Android, realizouse a configuración e a creación do proxecto sinxelamente.

6.4 Probas

Neste último apartado de probas realizáronse diferentes comprobacións para asegurar que a configuración e a posta a punto do proxecto foi correcta.

6.4.1 Servidor web

Nesta primeira iteración realizáronse probas usando a biblioteca JUnit, que nos permitiu realizar nun primeiro lugar probas unitarias aos repositorios que se describiron na sección

6.3.1. Grazas a éstas, comprobouse que se realizaban insercións correctas na base de datos e ao mesmo tempo que as anotacións de creacións das táboas no SGBD eran as correctas. Logo das probas unitarias realizáronse probas de integración, que nos permitiron probar todos os elementos unitarios que compoñen o sistema e que estes funcionaban correctamente probándoos todos xuntos.

6.4.2 Aplicación web - React

Nesta parte comprobouse que o proxecto quedou ben creado e era posible executalo sen ningún tipo de problema nun navegador.

6.4.3 Aplicación móbil - Android

Para esta parte comprobouse que se puido crear ben o proxecto Android e tamén lanzarse nun emulador para comprobar que o seu funcionamento era o correcto.

Iteración 2

Nesta iteración expoñerase a xestión de usuarios que nos permitirá para poder crealos, borrarlos ou editalos. Tamén se realizará a creación de animais, edición deles e borrado. Por último mostrarase a creación de tipos de animais e as razas asociadas a cada un deles.

7.1 Análise

A continuación, indicaranse os casos de uso respectivos a esta iteración. Cada un conterà un apartado de descrición, actor ou actores que interactúan con ese caso de uso e o fluxo básico para a súa execución no sistema. Agruparemos os casos de uso en diferentes subsistemas dependendo das funcionalidades que teñan. Nesta iteración mostrarase un esquema dos casos de uso da etapa de autenticación e rexistro, da creación edición e borrado de animais e da creación e borrado das razas e dos tipos de animais. Na figura 7.1 vemos de forma gráfica o reflectido nos casos de uso para este módulo.

7.1.1 Casos de uso - Xestión de usuarios, razas e tipos de animais

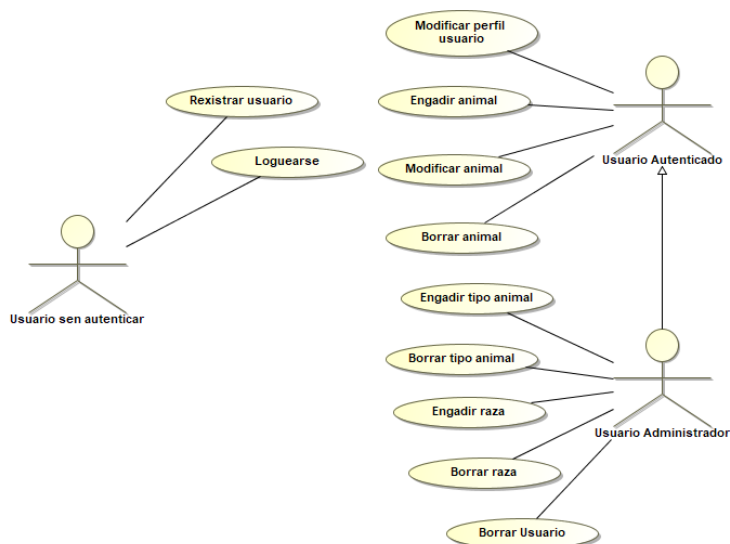


Figura 7.1: Casos de uso da iteración 2

CU-01: Rexistrar usuario

- **Actores** Usuario non rexistrado.
- **Descrición** Un usuario poderá rexistrarse na aplicación cubrindo diferentes datos que serán necesarios para outras operativas.
- **Fluxo básico**
 1. O usuario selecciona a opción Rexistrarse.
 2. O sistema mostra un formulario con opcións para completar.
 3. O usuario completa todos os campos seguintes: Nome, apelidos, email, dirección, código postal, teléfono e contrasinal.
 4. O sistema comprobará os datos e se son correctos e redixirá ao usuario a páxina de logueo.
- **Fluxo alternativo** 4a.Existe algún erro, infórmase ao usuario. Este terá que volver a cubrir os datos.

- **Precondicións** O usuario con mesmo email non pode rexistrarse dúas veces e o contrasinal debe ser de 8 ou máis caracteres.

- **Postcondicións** O usuario é creado.

CU-02: Loguearse un usuario

- **Actores** Usuario non rexistrado.
- **Descrición** Un usuario que xa esta rexistrado poderá ingresar na aplicación e ter as funcionalidades para un usuario autenticado.

- **Fluxo básico**

1. O usuario selecciona a opción Identificarse.
2. O sistema mostra un formulario con opcións para completar.
3. O usuario completa todos os campos email e contrasinal.
4. O sistema comprobará os datos e se son correctos o redixirá a páxina de anuncios.

- **Precondicións** O usuario ten que ten unha conta válida no sistema.
- **Postcondicións** O usuario quedará identificado no sistema e poderá facer uso das funcionalidades da aplicación para un usuario rexistrado.

CU-03: Modificar perfil de usuario

- **Actores** Usuario rexistrado e usuario administrador.
- **Descrición** Modificar os datos persoais de cada usuario.

- **Fluxo básico**

1. O usuario selecciona a opción Modificar perfil .
2. O sistema mostra un formulario con opcións a modificar.
3. O usuario completa os campos que quere editar, manténdose os mesmos datos en base de datos se non os cambia.
4. O sistema comprobará os datos e se son correctos, redixirá ao usuario a pantalla de ver anuncios.

- **Precondicións** Ter unha conta creada no sistema.
- **Postcondicións** Campos actualizados do perfil do usuario.

CU-04: Engadir animal

- **Actores** Usuario rexistrado e usuario administrador.
- **Descrición** O usuario rexistrado a través dun formulario poderá engadir animais.
- **Fluxo básico**
 1. O usuario administrador selecciona a opción ver animais.
 2. O sistema mostra unha táboa cos diferentes animais que posúe.
 3. O usuario seleccionará o botón engadir animal.
 4. O usuario completará un formulario para engadir o animal.
 5. O sistema comprobará os datos e se son correctos redirixirá ao usuario a pantalla dos animais.
- **Precondicións** O usuario ten que estar identificado.
- **Postcondicións** Queda creado o animal.

CU-05: Modificar animal

- **Actores** Usuario rexistrado e usuario administrador.
- **Descrición** Modificar os datos do animal.
- **Fluxo básico**
 1. O usuario administrador selecciona a opción ver animais.
 2. O sistema mostra unha táboa cos diferentes animais que posúe.
 3. O usuario seleccionará o botón editar animal en cada animal.
 4. O sistema comprobará os datos e se son correctos, redirixirá ao usuario a pantalla dos animais.
- **Precondicións** O usuario ten que estar identificado.
- **Postcondicións** Campos actualizados do animal.

CU-06: Borrar animal

- **Actores** Usuario rexistrado e usuario administrador.
- **Descrición** O usuario poderá borrar os animais xa existentes.
- **Fluxo básico**
 1. O usuario administrador selecciona a opción ver animais.

2. O sistema mostra unha táboa cos diferentes animais que posúe.
 3. O usuario seleccionará o botón borrar animal en cada animal.
 4. O sistema sacará un aviso buscando a confirmación final do usuario e procederá a borrar o animal.
- **Precondicións** O usuario ten que estar logueado.
 - **Postcondicións** O tipo de animal quedará borrado.

CU-07: Engadir tipos de animais

- **Actores** Usuario administrador.
- **Descrición** O usuario administrador a través dunha táboa poderá engadir tipos de animais.
- **Fluxo básico**
 1. O usuario administrador selecciona a opción editar razas e tipos de animais.
 2. O sistema mostra unha táboa cos diferentes tipos de animais que hai.
 3. O usuario engadirá o tipo de animal que desexe.
 4. O sistema comprobará os datos e se son correctos redirixirá ao usuario a pantalla de ver anuncios.
- **Precondicións** O usuario administrador ten que estar identificado.
- **Postcondicións** Queda creado o tipo de anuncio.

CU-08: Borrar tipo de animal

- **Actores** Usuario administrador.
- **Descrición** O usuario administrador a través dunha táboa poderá borrar os tipos de animais xa existentes.
- **Fluxo básico**
 1. O usuario administrador selecciona a opción editar razas e tipos de animais.
 2. O sistema mostra unha táboa cos diferentes tipos de animais que hai.
 3. O usuario eliminará o tipo de animal que desexe.
 4. O sistema sacará un aviso buscando a confirmación final do usuario e procederá a borrar o tipo de animal.

- **Precondicións** O usuario administrador ten que estar logueado.
- **Postcondicións** O tipo de animal quedará borrado.

CU-09: Engadir razas

- **Actores** Usuario administrador.
- **Descrición** O usuario administrador a través dunha táboa poderá engadir razas as xa existentes.
- **Fluxo básico**
 1. O usuario administrador selecciona a opción editar razas e tipos de animais.
 2. O sistema mostra unha táboa para poder engadir ou borrar razas de animais.
 3. O usuario engadirá unha raza as existentes seleccionando o tipo de animal ao que lla queira asignar.
 4. O sistema sacará un aviso buscando a confirmación final do usuario e procederá a crear a raza.

- **Precondicións** O usuario administrador debe estar identificado.
- **Postcondicións** Engadirase a razas creada polo administrador.

CU-10: Borrar razas

- **Actores** Usuario administrador.
- **Descrición** O usuario administrador a través dunha táboa poderá borrar a raza que desexa das existentes.
- **Fluxo básico**
 1. O usuario administrador selecciona a opción editar razas e tipos de animais.
 2. O sistema mostra unha táboa para poder borrar razas de animais.
 3. O usuario eliminará a raza que desexe.
 4. O sistema sacará un aviso buscando a confirmación final do usuario e procederá a crear a raza.

- **Precondicións** O usuario administrador debe estar identificado.
- **Postcondicións** Borrarase a raza seleccionada polo usuario.

CU-11: Borrar usuarios

- **Actores** Usuario administrador
- **Descripción** O usuario administrador a través dunha táboa poderá borrar usuarios os usuarios que desexe-
- **Fluxo básico**
 1. O usuario administrador selecciona a opción xestión de usuarios.
 2. O sistema mostra unha táboa cos usuarios rexistrados da aplicación.
 3. O usuario administrador eliminará a raza que desexe.
 4. O sistema sacará un aviso buscando a confirmación final do usuario e procederá a borrarlo.
- **Precondicións** O usuario administrador debe estar identificado.
- **Poscondicións** Borrarase o usuario seleccionado polo usuario adminsitrador, e en consecuencia, borrarase todo o creado polo usuario borrado na aplicación.

7.2 Deseño

Na fase de deseño desta iteración, deseñaremos os diagramas de clases que nos permitirán deixar claro o funcionamento interno da aplicación para esta iteración. Tamén poderemos ver os datos que nos fan falta para o funcionamento da mesma. No desenvolvemento desta iteración, como tamén nas restantes, farase especial énfase na separación dos diferentes módulos da aplicación: Servidor web, cliente web e cliente Android.

7.2.1 Deseño da aplicación - Servidor Web

Explicaremos nos seguintes apartados os aspectos máis relevantes do deseño de cada módulo da iteración.

- **Xestión da persistencia**

No diagrama de clases da Figura 7.2 podemos ver as entidades ou clases persistentes, que son as que se corresponderán coas diferentes entidades que almacenemos na base de datos. Cada clase ou entidade conterá, como mínimo, os mesmos atributos que a súa correspondente relacion en base de datos.

Para a abstracción destas entidades, como se indicou no apartado de patróns, sección 6.2.2, utilizamos o patrón Repository. Grazas ao uso do framework Spring Data JPA, as entidades do dominio do proxecto farán uso deste patrón extendendo dunha interfaz

que proporcionará operacións CRUD en un repositorio para cada tipo específico de entidade. Na Figura 7.3 vemos o uso deste patrón no noso servizo. Podemos ver como cada repositorio de cada entidade define unha serie de operacións para obter a información persistida da base de datos. Estas operacións serán traducidas por Spring a consultas SQL.

- **Lóxica de negocio. Servizos**

Para o desacoplamento do modelo do resto do sistema, tomouse a decisión da implementación de varias interfaces, seguindo o patrón de deseño Fachada, descrito na sección 6.2.2. En total, implementáronse 2 fachadas, unha para a xestión dos animais e outra para a xestión dos usuarios. Na figura 7.4 indicamos nun diagrama estes servizos. Podemos observar no diagrama o uso do patrón DTO, como se indicou na sección 6.2.2 de patróns, necesario para a comunicación entre os diferentes compoñentes da aplicación. necesitamos un elemento DTO. A continuación mostramos unha breve descripción dos servizos que vemos na imaxe 7.4.

- Animal Service Operacións relacionadas coa entidade Animal, coas razas e cos tipos de animais.
- Usuario Service Operacións relacionadas coa entidade Usuario.

- **Controlador Rest** O controlador Rest, permítenos xestionar as operacións chegadas dende o cliente ao servizo web, nesta iteración manexa a xestión de usuarios, a de animais e a das razas e tipos de animais.

7.2.2 Deseño da aplicación - Servizo Web

O deseño da parte web basease en compoñentes, que en esencia son unha peza da interface do usuario. A forma de estruturar estes compoñentes e que temos un compoñente raíz, este compoñente ten outros compoñentes "fillos" e estes a súa vez outros, etc. Na Figura 7.5 vemos un deseño dos compoñentes descritos nesta iteración.

O deseño da parte web basease en compoñentes, que en esencia son unha peza da interface do usuario. Na Figura 7.5 vemos os diferentes módulos nos que se organizaron os compoñentes, esta organización levouse a cabo según a relación que teñen uns compoñentes con outros. Así, por exemplo, no módulo *Usuario* estarían os compoñentes encargados de todas as operacións que teñan que ver cos usuarios do sistema.

Para esta iteración, realizáronse unha serie de maquetas que facilitaron en gran medida a implementación. Nas Figuras 7.6, 7.7, 7.7.7.7.7 e 7.7 podemos o formulario de rexistro, o de login, o de tipo de animais e o de razas, respectivamente.

7.3 Implementación

7.3.1 Servidor web

Como se expuxo no comezo deste capítulo, nesta iteración implementouse a lóxica de negocio para xestionar os usuarios e o manexo das razas e dos tipos de animais. A través do servizo creado para os usuarios, estableceuse un método que consiste no rexistro do usuario na base de datos e outro para poder identificarse na aplicación. Unha vez establecido o servizo creouse a denominada API REST, grazas tamén as notacións que nos proporciona o framework Spring. Isto permítenos que a través dunha chamada a esta API se chame ao servizo de rexistro de usuarios e, se os datos son correctos, estes se almacenen na base de datos, para logo, se o cliente o desexa, poder autenticarse chamando a outra operación da API e quedando autenticado. Este mesmo procedemento seguiuuse para a creación de razas e tipos de animais.

7.3.2 Aplicación web - React

Con respecto a implementación da parte web, como se explicou no apartado de deseño na sección web, [7.2.2](#), cada pantalla, cada compoñentes que hai que implementar, serán abstraídos e estarán independentes nunha carpeta, isto proporcionaranos moita máis claridade e simplificación do código.

7.4 Probas

7.4.1 Servidor web

Nesta última etapa de probas establecéronse probas unitarias para a parte do servidor, nela probáronse a través de JUnit as diferentes funcionalidades implementadas, como foron a xestión do rexistro de usuarios, logueo dun usuario ou creación dun animal por parte dun usuario. Grazas a estas probas puidemos comprobar que se realizaba a creación de usuarios, de razas ou de tipos de animais. Logo realizáronse probas de integración que nos permitiron probar o correcto funcionamento de todos os compoñentes implementados ata agora.

7.4.2 Aplicación web - React

Na parte web realizáronse probas de usabilidade da aplicación para ver que o funcionamento era o correcto e se estaban realizando as operacións correctamente.

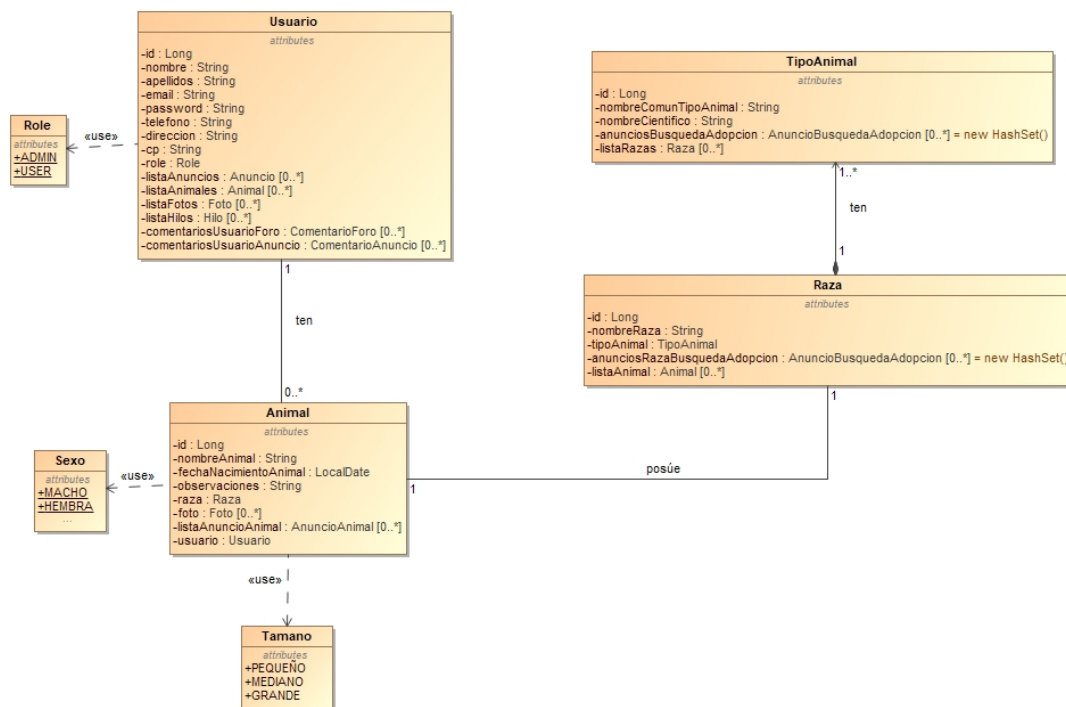


Figura 7.2: Diagrama de clases da iteración 2

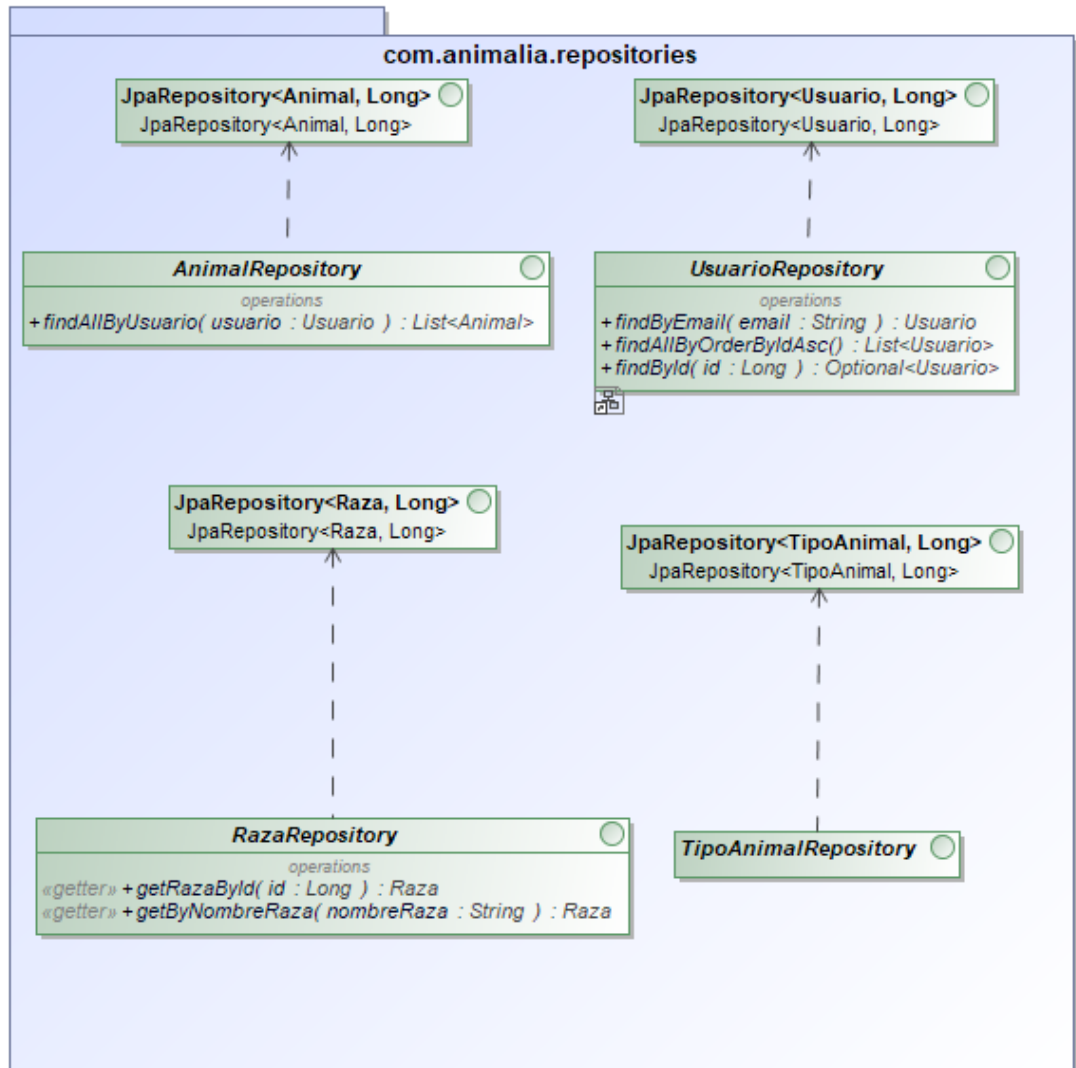


Figura 7.3: Entidades do dominio da iteración 2. Interface repositorios

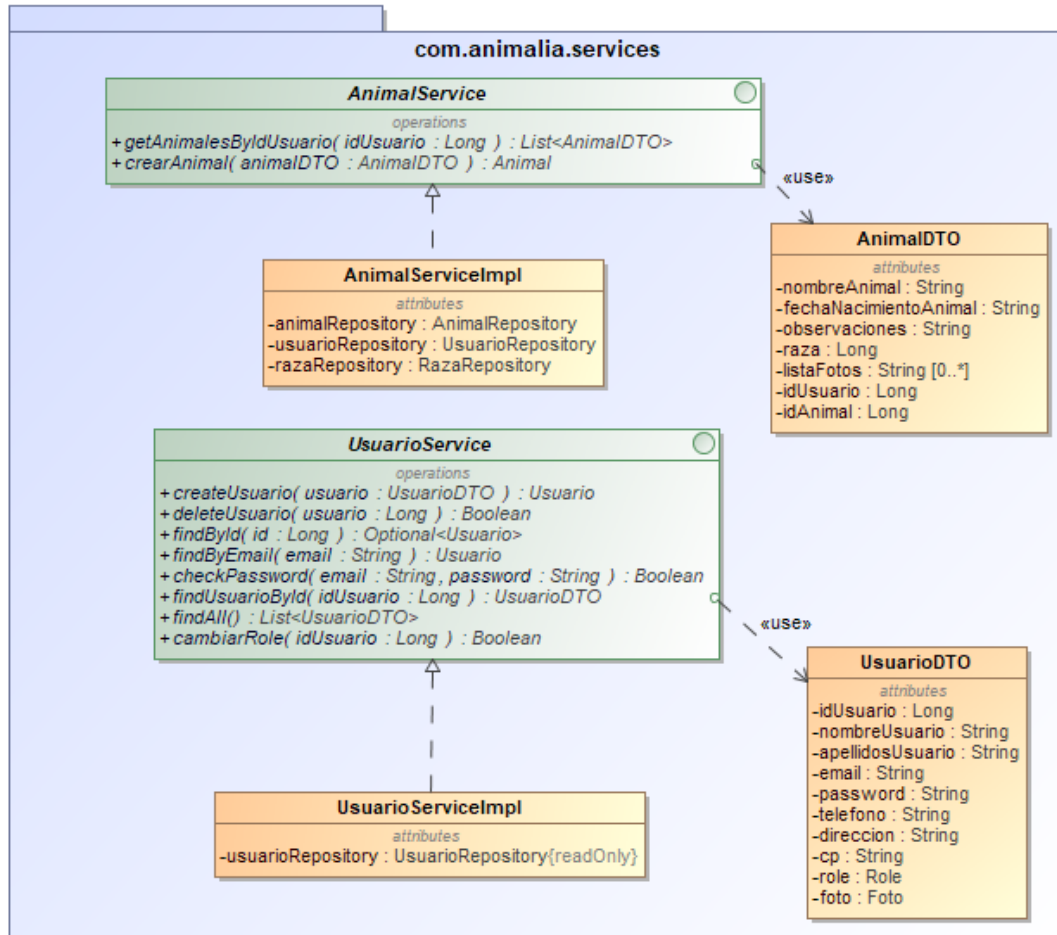


Figura 7.4: Diagrama dos servizos da iteración 2

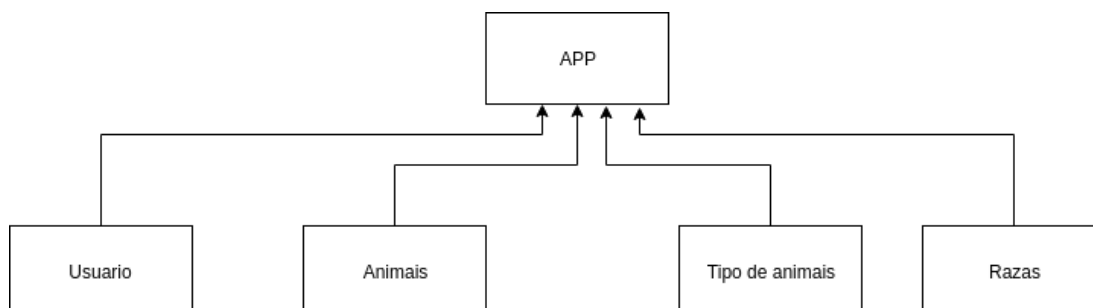


Figura 7.5: Diagrama dos compoñentes usados no servizo web na iteración 2

Maqueta de un formulario de registro en la aplicación web Animalia. El formulario se presenta en un navegador web con una barra de direcciones y botones de navegación. El título principal es "Bienvenido a Animalia". El formulario contiene los siguientes campos:

Nombre	<input type="text"/>	Apellidos	<input type="text"/>	Email	<input type="text"/>	Dirección	<input type="text"/>
CP	<input type="text"/>	Teléfono	<input type="text"/>	Contraseña	<input type="password"/>	Repetir contraseña	<input type="password"/>

Debajo de los campos, hay un botón "Guardar usuario".

Figura 7.6: Maqueta do formulario de rexistro na aplicación web

The image shows a wireframe of a web browser window. The browser's address bar is empty. Below the address bar, there is a line of text: "Ítem Integración con otras plataformas de anuncios. Sería de gran utilidad que otras plataformas se integraran con la aplicación permitiendo la creación de anuncios". In the center of the page is a "Log in" form. The form has a title "Log in" at the top. Below the title, there are two input fields: "Email" and "Password". The "Email" field contains the text "Email". The "Password" field contains the text "*****". Below the "Password" field, there is a checkbox labeled "Remember me" which is checked. Below the checkbox, there are two buttons: "Log in" and "Registrarse".

Ítem Integración con otras plataformas de anuncios. Sería de gran utilidad que otras plataformas se integraran con la aplicación permitiendo la creación de anuncios

Log in

Email Email

Password *****

☒ Remember me

Log in

Registrarse

Figura 7.7: Maqueta do formulario de login na aplicación web

La imagen muestra una maqueta de una interfaz web dentro de un navegador. El navegador tiene una barra de direcciones vacía y una barra de título que dice "Bienvenido a Animalia, Nombre de usuario". En la parte superior derecha del navegador hay enlaces a "Mis anuncios", "Foro" y "Salir", junto con un icono de configuración.

El contenido principal de la página es un formulario titulado "Gestión de tipo de animales". El formulario incluye un campo de texto con el placeholder "escribir tipo animal..." y un botón "Guardar".

Debajo del campo de texto hay una tabla con dos columnas: "Nombre tipo animal" y "Borrar tipo animal". La tabla contiene dos filas de datos.

Nombre tipo animal	Borrar tipo animal
Nombre tipo animal 1	
Nombre tipo animal 2	

Figura 7.8: Maqueta do formulario da xestión de tipos de animais da aplicación web

The wireframe shows a web browser window with a title bar, address bar, and navigation buttons. The page content includes a welcome message, user navigation links, a main title, a form to add a breed, and a table of existing breeds with delete buttons.

Bienvenido a Animalia, Nombre de usuario

Mis anuncios Foro Salir

Gestión de razas

escribir raza... Guardar

Nombre tipo animal	Borrar tipo animal
Nombre raza 1	
Nombre raza 2	

Figura 7.9: Maqueta do formulario da xestión de razas da aplicación web

Nesta iteración expoñerase o estudo sobre a creación, actualización e borrado de anuncios de animais por parte dun usuario, o poder ver o listaxe dos seus anuncios e doutros usuarios e xestionar os envíos de emails para posibles adopcións.

8.1 Análise

Nesta iteración mostrarase un esquema dos casos de uso da etapa de creacion e borrado dun anuncio dependendo do tipo. Tamén se xestionará os posibles comentarios nos diversos anuncios. Vemos o diagrama de casos de uso asociado a esta iteración na Figura 8.1.

8.1.1 Casos de uso - Xestión de anuncios

CU-12: Crear anuncio de desaparición

- **Actores** Usuario rexistrado e administrador
- **Descrición** Un usuario poderá crear un anuncio sobre unha desaparición.
- **Fluxo básico**
 1. O usuario selecciona a opción crear anuncio de animal desaparecido.
 2. O sistema mostra un formulario coas opcións para completar.
 3. O usuario completa os campos seguintes: Título do anuncio, descrición do anuncio, tipo de animal, raza do animal, data de nacemento, observacións, nome do animal, localización da desaparición do animal e foto do animal desaparecido.
 4. O sistema comprobará os datos, e se son correctos redirixirá ao usuario a páxina de anuncios xa con ese anuncio creado.

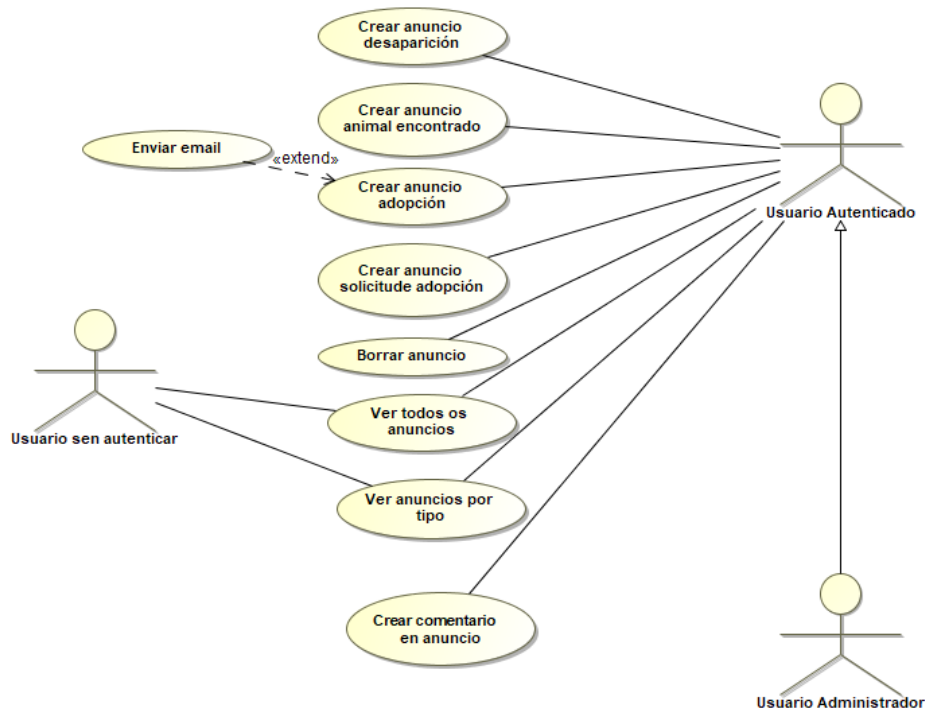


Figura 8.1: Casos de uso da iteración 3

- **Fluxo alternativo** 4a.Existe algún erro, infórmase ao usuario. Este terá que volver a cubrir os datos.
- **Precondicións** O usuario ten que estar logueado.
- **Poscondicións** O anuncio e o animal asociado quedan creados na base de datos.

CU-13: Crear anuncio de animal encontrado

- **Actores** Usuario rexistrado e administrador
- **Descrición** Un usuario poderá crear un anuncio sobre un animal encontrado.
- **Fluxo básico**
 1. O usuario selecciona a opción crear anuncio de animal encontrado.
 2. O sistema mostra un formulario coas opcións para completar.
 3. O usuario completa os campos seguintes: Título do anuncio, raza e tipo de animal, descrición do anuncio, localización da aparición e foto do animal encontrado.

4. O sistema comprobará os datos e se son correctos redirixirá ao usuario a páxina de anuncios xa con ese anuncio creado.
- **Fluxo alternativo** 4a.Existe algún erro, infórmase ao usuario. Este terá que volver a cubrir os datos.
 - **Precondicións** O usuario ten que estar logueado.
 - **Poscondicións** O anuncio queda creado.

CU-14: Crear anuncio de adopción

- **Actores** Usuario rexistrado e administrador
- **Descrición** Un usuario poderá crear un anuncio sobre unha adopción.
- **Fluxo básico**
 1. O usuario selecciona a opción crear anuncio de adopción.
 2. O sistema mostra un formulario coas opcións para completar.
 3. O usuario completa os campos seguintes: Título do anuncio, raza e tipo de animal, descrición do anuncio, localización da aparición e foto do animal que se pon en adopción.
 4. O sistema comprobará os datos e se son correctos redirixirá ao usuario a páxina de anuncios xa con ese anuncio creado.
- **Fluxo alternativo** 4a.Existe algún erro, infórmase ao usuario. Este terá que volver a cubrir os datos. 4b.No caso de existir un anuncio de solicitude de adopción cuxos requisitos sexan un dos creados neste anuncio enviarase un email de aviso ao usuario que creou dito anuncio.
- **Precondicións** O usuario ten que estar logueado.
- **Poscondicións** O anuncio queda creado.

CU-15: Crear anuncio de solicitude de adopción

- **Actores** Usuario rexistrado e administrador
- **Descrición** Un usuario poderá crear un anuncio sobre unha solicitude de adopción.
- **Fluxo básico**
 1. O usuario selecciona a opción crear solicitude de adopción.

2. O sistema mostra un formulario coas opcións para completar.
 3. O usuario completa os campos seguintes: Título do anuncio, raza e tipo de animal, sexo, tamaño e idade do animal, descrición do anuncio e localización de onde vive o usuario.
 4. O sistema comprobará os datos e se son correctos redirixirá ao usuario a páxina de anuncios xa con ese anuncio creado.
- **Fluxo alternativo** 4a.Existe algún erro, infórmase ao usuario. Este terá que volver a cubrir os datos.
 - **Precondicións** O usuario ten que estar logueado.
 - **Poscondicións** O anuncio queda creado.

CU-16: Borrado dun anuncio

- **Actores** Usuario rexistrado e administrador
- **Descrición** Un usuario poderá borrar un anuncio creado anteriormente por él.
- **Fluxo básico**
 1. O usuario selecciona a opción borrar anuncio na táboa de meus anuncios.
 2. O sistema saca un aviso para confirmar que se desexa borrar ese anuncio.
 3. O usuario confirma o borrado do anuncio e este é borrado.
 4. O sistema comprobará os datos e se son correctos redirixirá ao
- **Fluxo alternativo** 4a.Si existe algún erro, infórmase ao usuario.
- **Precondicións** O usuario ten que estar logueado.
- **Poscondicións** O anuncio queda borrado.

CU-17: Ver todos os anuncios

- **Actores** Usuario sen rexistrar
- **Descrición** Un usuario poderá consultar todos os anuncios que existen.
- **Fluxo básico**
 1. O usuario entra na aplicación e será redirixido a páxina principal para ver todos os anuncios.
 2. O sistema mostrará todos os anuncios creados na aplicación.

- **Fluxo alternativo** 4a.Si existe algún erro, infórmase ao usuario.

- **Precondicións**

- **Poscondicións**

CU-18: Ver anuncios dependendo do tipo

- **Actores** Usuario sen rexistrar

- **Descrición** Un usuario poderá consultar todos os anuncios que existen dependendo do tipo de anuncio

- **Fluxo básico**

1. O usuario entra na aplicación e será redirixido a páxina principal para ver todos os anuncios.
2. O sistema mostrará todos os anuncios creados na aplicación.
3. O usuario seleccionará calquera dos catro tipos que existen e será redirixido a unha páxina onde se mostrarán os anuncios dese tipo

- **Fluxo alternativo** 4a.Si existe algún erro, infórmase ao usuario.

- **Precondicións**

- **Poscondicións** Móstranse os anuncios dependendo do tipo de anuncio seleccionado.

CU-19: Crear comentario nun anuncio

- **Actores** Usuario rexistrado ou administrador

- **Descrición** Un usuario poderá poñer un comentario en calquera anuncio.

- **Fluxo básico**

1. O usuario entra nun anuncio en particular.
2. O sistema mostrará os datos dese anuncio.
3. O usuario escribirá un comentario no anuncio
4. O usuario confirmará o comentario e este será creado.

- **Fluxo alternativo** 4a.Si existe algún erro, infórmase ao usuario.

- **Precondicións** O comentario non pode ir vacío.

- **Poscondicións** Crease o comentario e queda visible no anuncio.

8.2 Deseño

No desenvolvemento desta iteración, como tamén nas restantes, farase especial énfase na separación dos diferentes módulos da aplicación: Servidor web, cliente web e cliente Android.

8.2.1 Deseño da aplicación - Servidor web

- **Xestión da persistencia**

No diagrama de clases da Figura 8.2 podemos ver as entidades ou clases persistentes. Como se aprecia no diagrama de clases, vemos o deseño desta iteración e da anterior.

- **Lóxica de negocio. Servizos**

Para o desacoplamento do modelo do resto do sistema, tomouse a decisión da implementación de varias interfaces, seguindo o patrón de deseño Fachada, descrito na sección 6.2.2. En total, nesta iteración implementáronse 3 fachadas, unha para a xestión das fotos, outra para a xestión dos anuncios e outra para os envíos de emails. Na Figura 8.3 indicamos nun diagrama estes servizos. Podemos observar no diagrama o uso do patrón DTO, como se indicou na sección 6.2.2 de patróns, necesario para a comunicación entre os diferentes compoñentes da aplicación. necesitamos un elemento DTO.

- **Controlador Rest** O controlador Rest permítenos xestionar as operacións chegadas dende o cliente ao servizo web, nesta iteración manexa a xestión de anuncios e os comentarios neles.

8.2.2 Deseño da aplicación - Servizo Web

No que respecta a parte web, na Figura 8.4 vemos os compoñentes para a aplicación web engadidos nesta iteración.

Nesta iteración realizáronse tamén as maquetas das diferentes interfaces, para que resultara moito mais sinxelo a súa implementación nun futuro. Nas Figuras 8.5, 8.6, 8.7 e 8.7 podemos observar a listaxe de todos os anuncios anuncios, a vista dun anuncio particular e unha lista de anuncios filtrada por o seu tipo, respectivamente.

8.2.3 Deseño da aplicación - Cliente Móbil

No que respecta a parte móbil, deseñouse unha fachada que nos permite a xestión dos anuncios no cliente móbil. Vemos este exemplo na Figura .

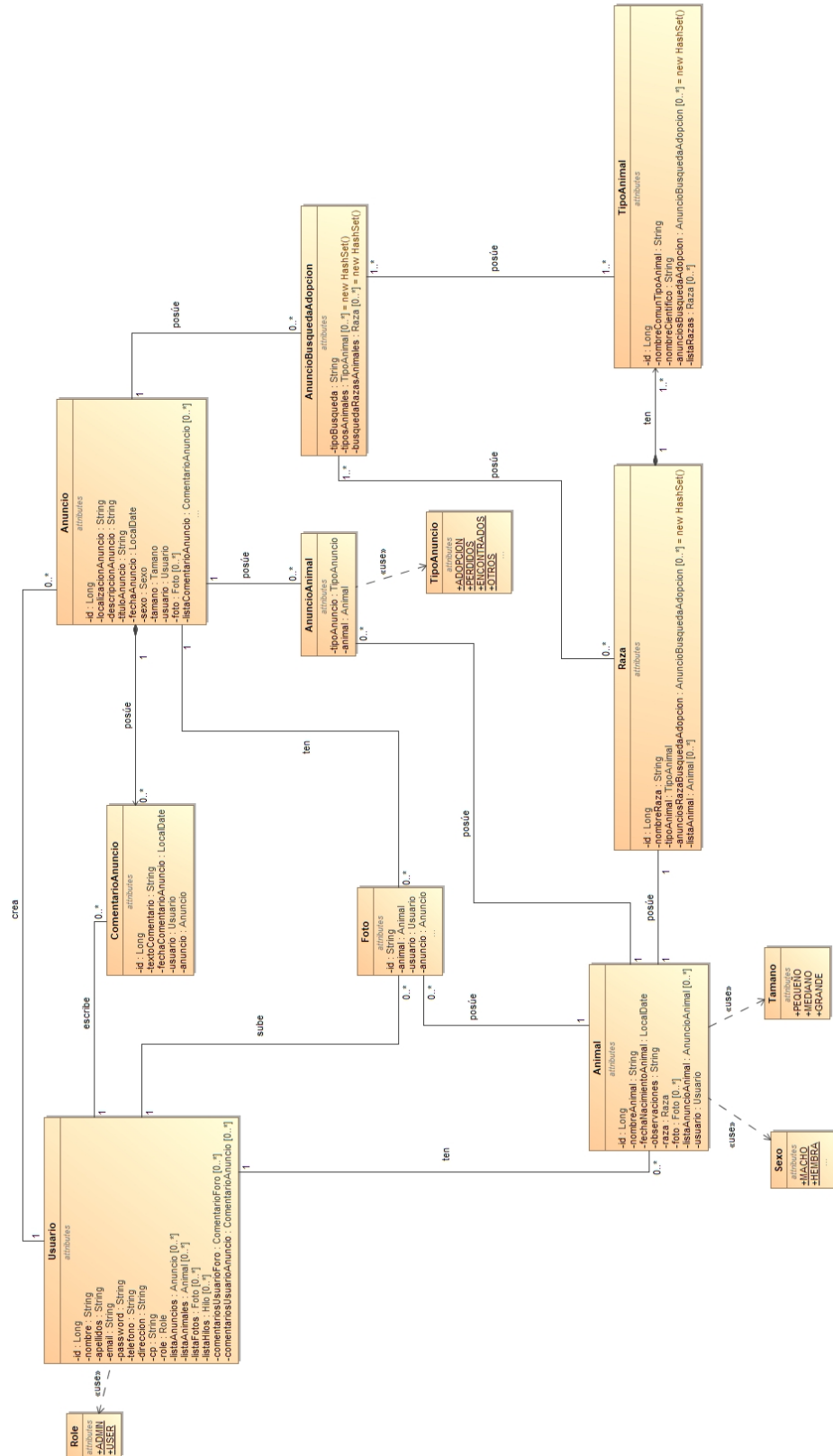


Figura 8.2: Diagrama de clases da iteración 3

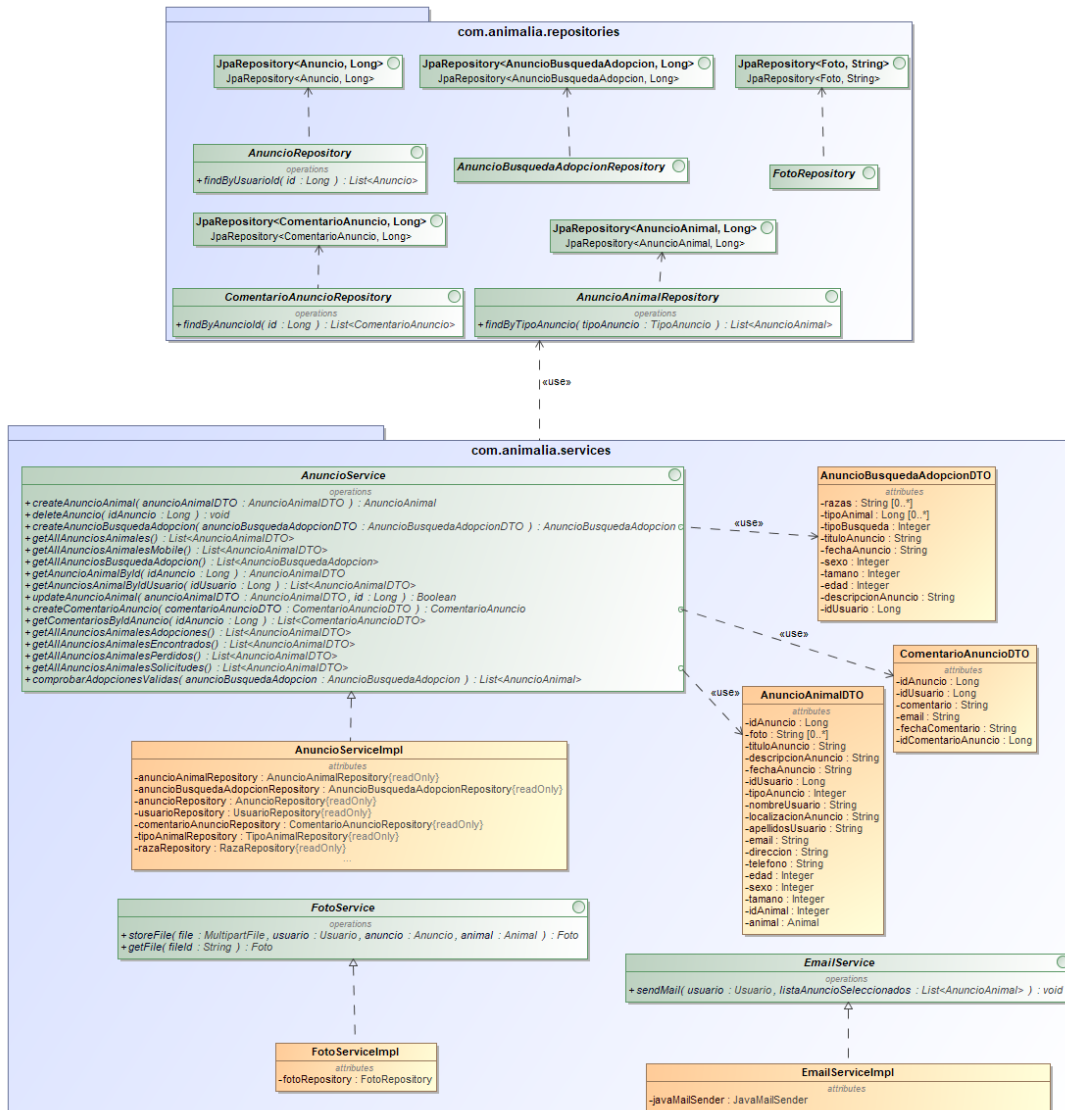


Figura 8.3: Diagrama dos servizos da iteración 3

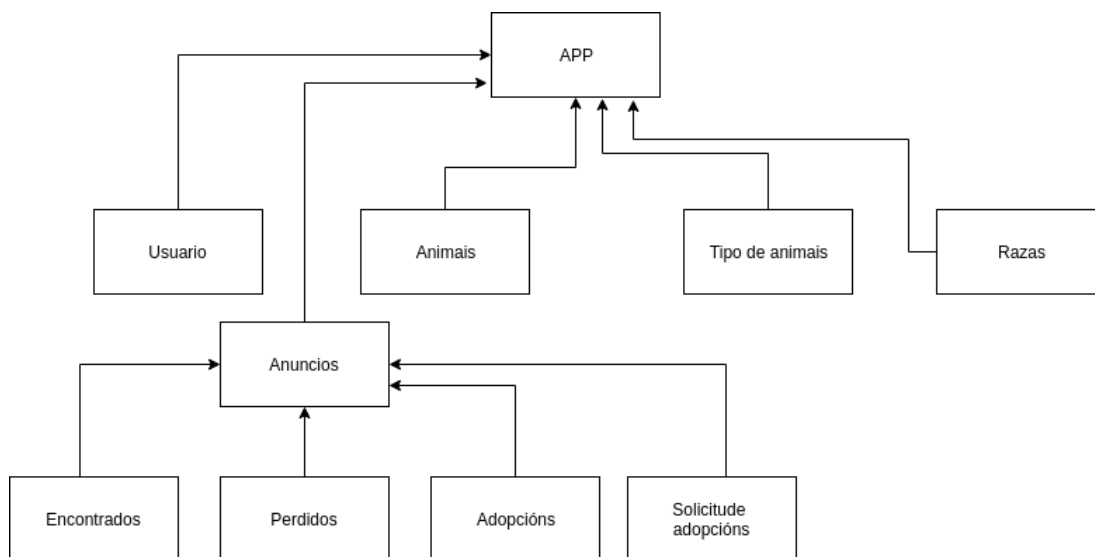


Figura 8.4: Diagrama dos compoñentes usados no servizo web na iteración 3



Figura 8.5: Maqueta do listado de anuncios da aplicación web

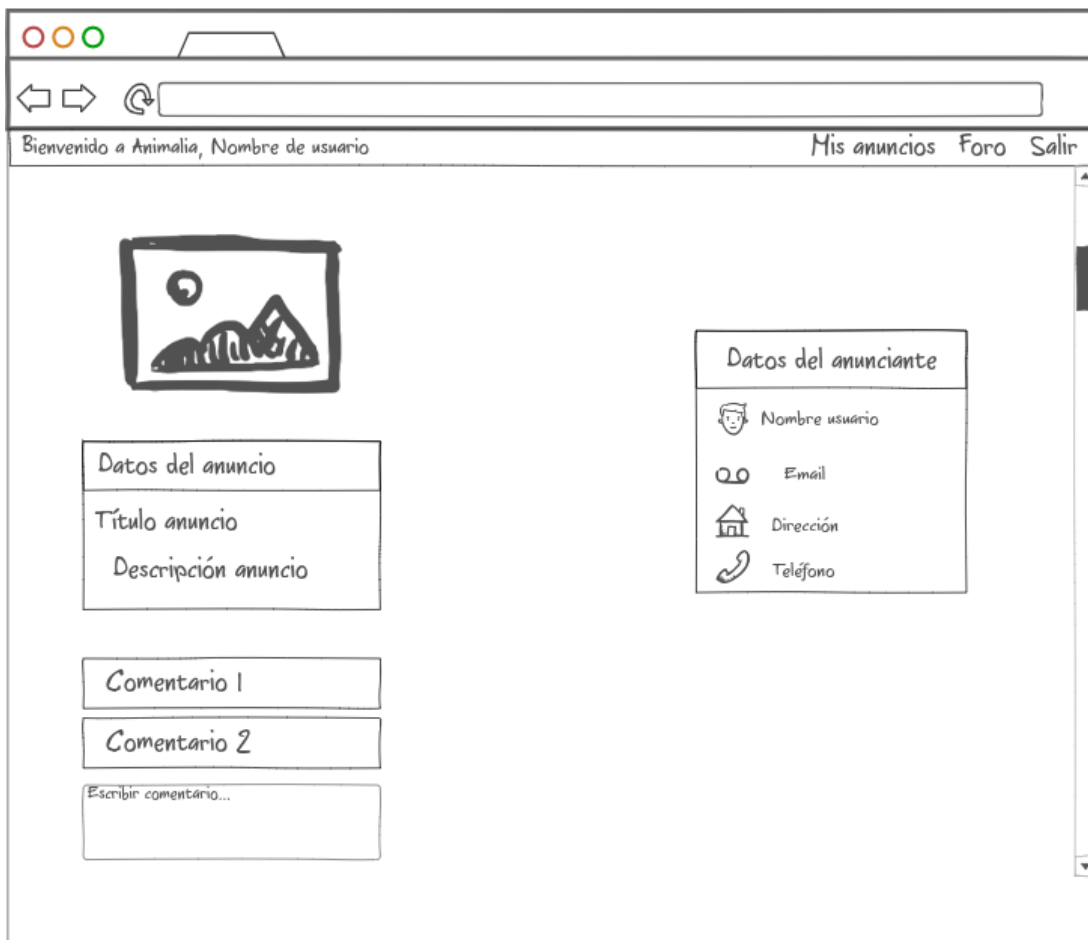


Figura 8.6: Maqueta dun anuncio particular da aplicación web



Figura 8.7: Maqueta da lista de anuncios filtrados por un tipo en particular

8.3 Implementación

8.3.1 Servidor web

Nesta fase implementouse a xestión dos anuncios, realizáronse os servizos que corresponden coa lóxica de negocio dos anuncios e a súa integración cos repositorios. Unha vez realizados estes servizos, implementouse a API REST para poder chamar a estes e que o cliente poida facer uso destas operativas.

8.3.2 Aplicacion web - React

No que refire a aplicación web, implementáronse as pantallas que ver todos os anuncios e os anuncios en particular, e tamén a creación de anuncios.

8.3.3 Aplicacion móbil - Android

Por último, no que se refire a aplicación Android, implementouse a pantalla inicial, que e a de vista de todos os anuncios e logo implementáronse a pantalla na que se pode ver un anuncio en particular.

8.4 Probas

8.4.1 Servidor web

Para a parte do servidor web fixéronse probas usando a biblioteca JUnit. Grazas a esta librería realizáronse probas unitarias dos diferentes servizos creados, algúns casos probados foron crear anuncios, borrar, envío de emails...Logo de rematar as probas unitarias realizáronse as probas de integración que nos permitiron probar as funcionalidades implementadas todas xuntas.

8.4.2 Aplicacion web - React

Na parte web realizáronse probas de uso, nas que se foi probando o sistema e vendo se o funcionamento era o correcto e non se producían erros ao crear anuncios, borrar anuncios, enviar emails... nin na aplicación web nin en consecuencia na parte do servidor.

8.4.3 Aplicacion móbil - Android

Na parte Android tamén se realizaron probas de uso, certificando que os anuncios se mostraban correctamente na interface e que as diferentes transicións da aplicación móbil o funcionamento era o correcto.

Iteración 4

Nesta última iteración implantouse a xestión do foro e as súas operacións asociadas.

9.1 Análise

Nesta etapa de análise indicaremos os casos de uso que afectan ao foro.

9.1.1 Casos de uso - Xestión do foro

Na figura 9.1 vemos un diagrama dos casos de uso que afectan a esta iteración.

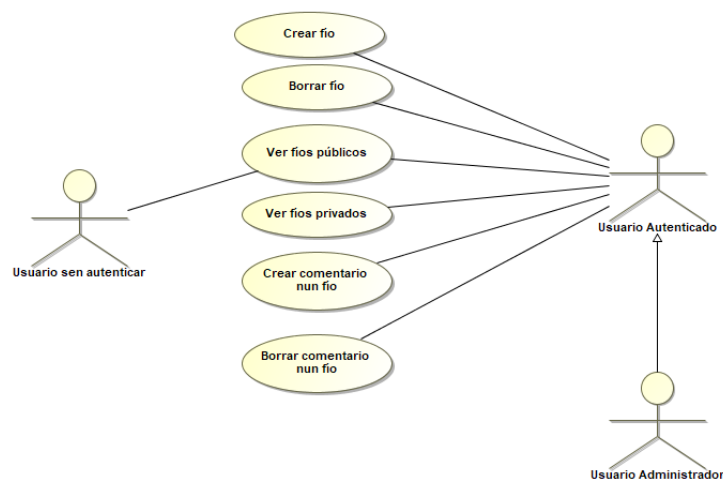


Figura 9.1: Casos de uso da iteración 4

CU-20: Crear fio

- **Actores** Usuario rexistrado e administrador.
- **Descrición** Un usuario poderá crear un fío no foro.
- **Fluxo básico**
 1. O usuario selecciona a opción crear fío.
 2. O sistema mostra un formulario coas opcións para completar.
 3. O usuario completa os campos seguintes: Título do fío, texto do fío, temática e visibilidade do fío.
 4. O sistema comprobará os datos, e se son correctos creará o fío.
- **Fluxo alternativo** 4a.Existe algún erro, infórmase ao usuario. Este terá que volver a cubrir os datos.
- **Precondicións** O usuario ten que estar logueado.
- **Poscondicións** O fío quedará creado.

CU-21: Ver fíos públicos

- **Actores** Usuario sen rexistro.
- **Descrición** Un usuario sen rexistrar poderá consultar os fíos públicos.
- **Fluxo básico**
 1. O usuario selecciona a opción ver fíos.
 2. O sistema mostra os fíos de visibilidade pública.
- **Precondicións**
- **Poscondicións**

CU-22: Ver fíos privados

- **Actores** Usuario rexistrado
- **Descrición** Un usuario rexistrado poderá consultar todos os fíos.
- **Fluxo básico**
 1. O usuario selecciona a opción ver fíos.
 2. O sistema mostra todos os fíos.

- **Precondicións**

- **Poscondicións**

CU-23: Crear comentario nun fío

- **Actores** Usuario rexistrado e administrador.

- **Descrición** Un usuario poderá comentar nun fío.

- **Fluxo básico**

1. O usuario selecciona un fío na lista de fíos.
2. O sistema mostra un fío.
3. O usuario escribe un comentario na sección de comentarios.
4. O sistema comprobará os datos, e se son correctos creará o comentario asociado ao fío.

- **Fluxo alternativo** 4a.Existe algún erro, infórmase ao usuario. Este terá que volver a cubrir os datos.

- **Precondicións** O usuario ten que estar logueado.

- **Poscondicións** O comentario quedará creado.

CU-24: Borrar comentario

- **Actores** Usuario rexistrado e administrador.

- **Descrición** Un usuario poderá borrar un comentario creado por él.

- **Fluxo básico**

1. O usuario selecciona un fío na lista de fíos.
2. O sistema mostra un fío.
3. O usuario borra o comentario que creou.

- **Fluxo alternativo** 3a.Existe algún erro, infórmase ao usuario. Este terá que volver a cubrir os datos.

- **Precondicións** O usuario ten que estar logueado.

- **Poscondicións** O comentario queda borrado.

CU-25: Borrar fío

- **Actores** Usuario rexistrado e administrador.
- **Descrición** Un usuario poderá borrar fío e os comentarios asociados a él.
- **Fluxo básico**
 1. O usuario selecciona un fío na lista de fíos.
 2. O sistema mostra un fío.
 3. O usuario borra o fío que creou.
- **Fluxo alternativo** 3a.Existe algún erro, infórmase ao usuario. Este terá que volver a cubrir os datos.
- **Precondicións** O usuario ten que estar logueado.
- **Poscondicións** O fío e os comentarios dese fío quedan borrados.

9.2 Deseño

No desenvolvemento desta iteración, como tamén nas restantes, farase especial énfase na separación dos diferentes módulos da aplicación: Servidor web, cliente web e cliente Android.

- **Xestión da persistencia** No diagrama de clases da Figura 6.4 podemos apreciar o deseño da última iteración integrado a xestión do foro e os seus comentarios coas outras iteracións xa realizadas.
- **Lóxica de negocio. Servizos** Na Figura 9.2 como nas outras iteracións decidiuse a implementacion do patrón fachada, neste caso implementouse unha soa, que sería a necesaria para a xestión do forn.
- **Controlador Rest** O controlador Rest, permítenos xestionar as operacións chegadas dende o cliente ao servizo web, nesta iteración manéxase a xestión do foro, procesa a creación de fíos, a creacion de comentarios o borrado de estes...

9.2.1 Deseño da aplicación - Servizo Web

O deseño da parte web basease en compoñentes, que en esencia son unha peza da interface do usuario. A forma de estruturar estes compoñentes e que temos un compoñente raíz, este compoñente ten outros compoñentes "fillos" e estes a súa vez outros, etc. Na Figura 9.3 vemos un deseño dos compoñentes descritos nesta iteración.

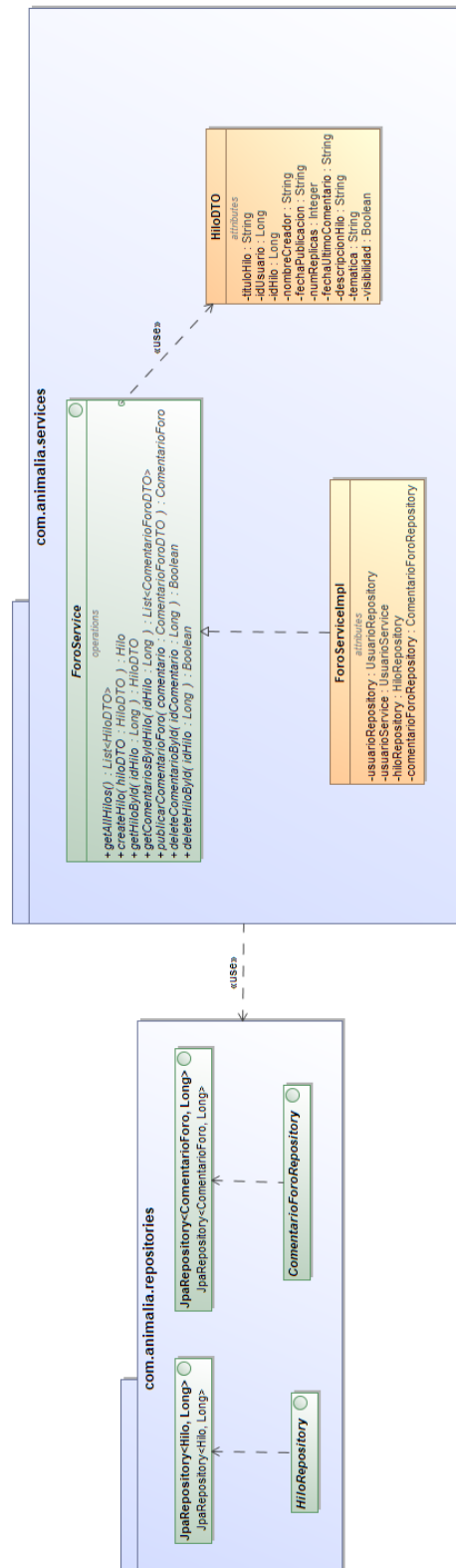


Figura 9.2: Diagrama dos servizos da iteración 4

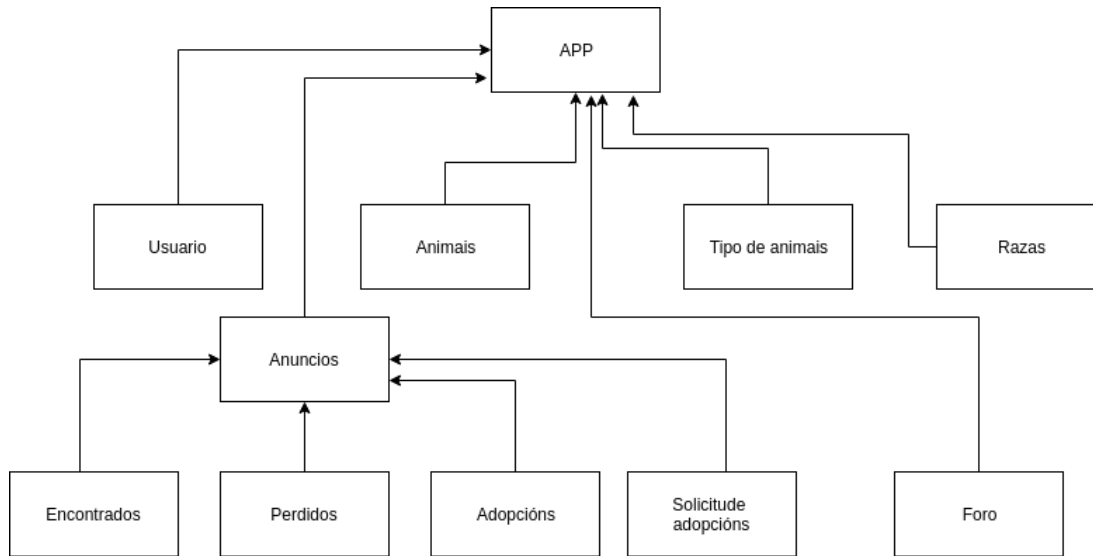


Figura 9.3: Diagrama dos compoñentes usados no servizo web na iteración 4

Nesta iteración realizáronse diferentes maquetas para un deseño inicial das pantallas. Nas Figuras 9.4 e 9.5 indícanse a lista de fíos e a vista dun fío en particular.

9.2.2 Deseño da aplicación - Cliente Móbil

No que respecta a parte móbil, deseñouse unha fachada que nos permite a xestión do foro no cliente móbil.

9.3 Implementación

9.3.1 Servidor web

No que refire a implementación do servidor nesta iteración, fíxose a creación dos servizos que necesitamos para o manexo do foro, como foron a creación dun fío, a dos comentarios asociados a un fío, a do borrado de comentarios e tamén a funcionalidade de mostrar todos os fíos que foron creados.

9.3.2 Aplicación web - React

No que decante a aplicación web, implementáronse as pantallas para o manexo do foro, creando unha para ver todos fíos dispoñibles, outra para ver un fío en concreto e os seus comentarios asociados e outra para poder crear os fíos.



Figura 9.4: Maqueta dos diferentes fíos do foro que temos na aplicación

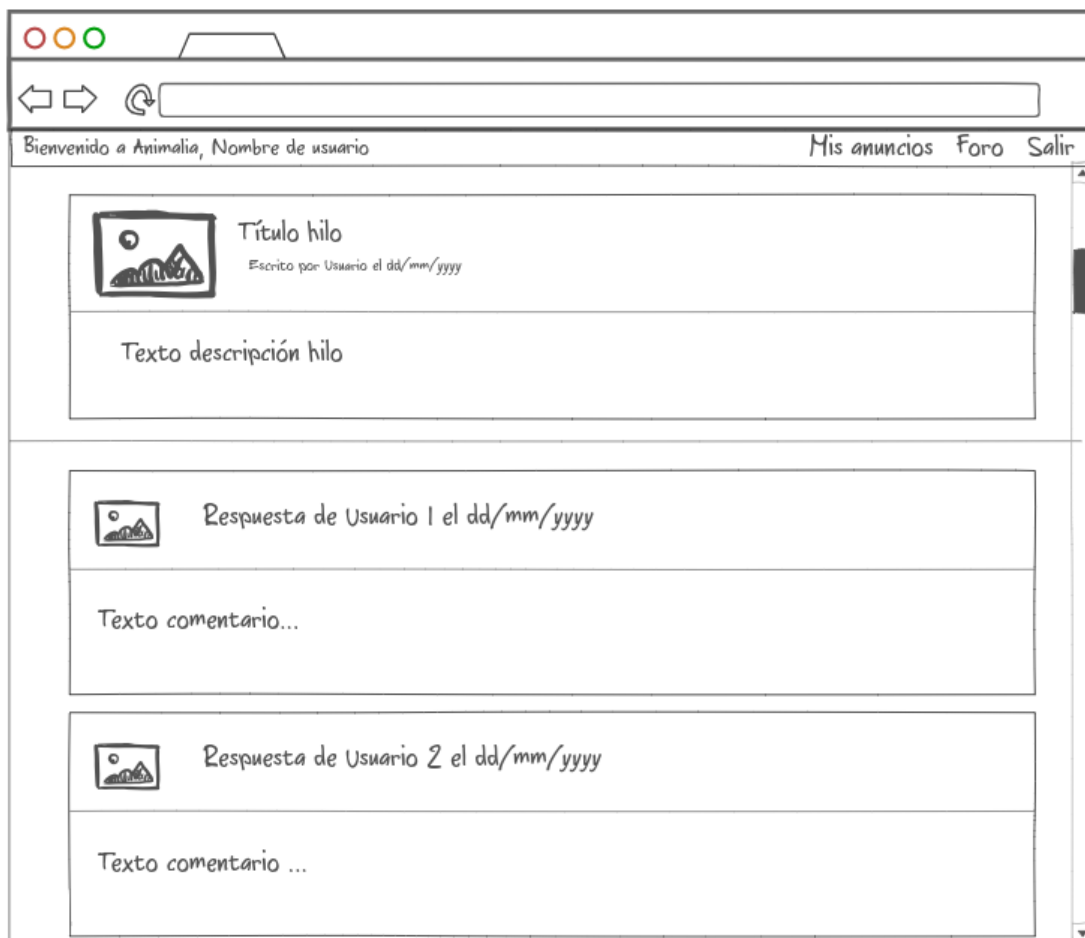


Figura 9.5: Maqueta dunha interface que representa a vista dun fío do foro na aplicación

9.3.3 Aplicación móvil - Android

No que se refire a aplicación Android fíxose a implementación das pantallas que xestionan o foro, para poder usar as funcionalidades que foron comentadas anteriormente.

9.4 Probas

9.4.1 Servidor web

Realizáronse probas de unidade coa biblioteca JUnit nos servizos implementados. Intentáronse buscar erros que poideran ocorrer ao crear un fío ou a borrar os comentarios. Logo destas probas realizáronse probas de integración para probar a execución correcta de todas as partes implementadas.

9.4.2 Aplicación web - React

Para a parte web, realizáronse probas de uso na que a través das funcionalidades implementadas, simulouse a creacións e borrado de fíos intentando buscar posibles erros na implementación.

9.4.3 Aplicación móvil - Android

Por último, intentouse facer o mesmo que na aplicación web, a través de probas de uso comprobouse que o funcionamento era o correcto e que non se producía ningún erro ao intentar mostrar as funcionalidades do foro implementadas.

Conclusións e traballo futuro

10.1 Conclusións

O produto final cumpre cos obxectivos explicados especificados na introdución 1:

- Dispónse dun sistema de creación de anuncios de catro tipos diferentes
- A aplicación conta con un sistema de comentarios nos anuncios para que os usuarios rexistrados poidan interactuar co creador do anuncio
- A aplicación conta un sistema de alertas para a xestión de adopcións, o sistema enviará un email a un usuario se aparece algún anuncio que cumpra os seus desexos de adopción
- Dispón tamén dun pequeno foro para comentar novas de diferentes tipos sobre animais
- Desenvolveuse unha aplicación web que permite consultar todos os anuncios rapidamente creados polos usuarios na páxina web
- A aplicación móbil permitirá consultar o foro para resolver dúbidas de forma rápida

O desenvolvemento do proxecto abarcou os requisitos inicialmente estipulados. Desenvolveuse unha ferramenta que actualmente non hai no mercado, que integra un foro e a creación de diferentes tipos de anuncios, por exemplo, a desaparición dun animal ou a chegada dun.

Con respecto a obtención de coñecementos, obtivéronse capacidades no manexo de Android, de React e de JavaScript, que son tecnoloxías moi importantes no día de hoxe e que serán de gran axuda nun futuro. Por último destacar tamén a consecución dunha aplicación con diversos compoñentes e conseguir que todos se poidan integrarse facilmente.

10.2 Traballo futuro

Aparte dos requisitos estipulados no punto 1.2 podemos engadir os seguintes para futuras melloras do sistema:

- Internacionalización da aplicación para que poida soportar mais idiomas que o español.
- Integración con outras plataformas de anuncios. Sería de gran utilidade que outras plataformas se integraran coa aplicación permitindo a creación de anuncios fora do noso portal. Un exemplo seria con Facebook.
- Creacion dun menú de login e rexistro para os usuarios na aplicación móvil. Co uso das tecnoloxías móbiles hoxe en día, sería práctico que a aplicación móbil implementara a administración de anuncios e tamén que estes usuarios poideran crear fíos a través do móbil.
- Creación da aplicación iOS. Unha aplicación para sistemas Apple que permitira tamén a consulta dos anuncios a través dunha aplicación nativa.
- Diseñar un sistema de recompensas para quen atope un animal. Se unha persoa encontra unha mascota e avisa ao dono, que este poida ser recompensado, grazas a un sistema de recompensas que poida integrar a aplicación.
- Engadir un sistema de xestión para protectoras de animais. Que as protectoras poidan ter un manexo mais fácil da creación de anuncios en masa así como engadir todo o necesario para unha xestión integral dos seus procesos internos.

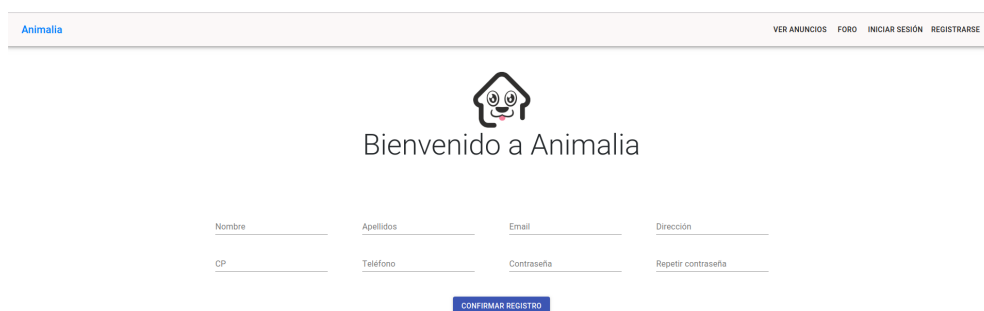
Apéndices

Manual de usuario

A continuación indicaremos a través de diferentes figuras un pequeno manual de usuario para presentar as aplicacións desenvolvidas.

Aplicación web

A través deste formulario de rexistro, podemos crear unha conta na aplicación cubrindo os datos solicitados:



Animalia VER ANUNCIOS FORO INICIAR SESIÓN REGISTRARSE

Bienvenido a Animalia

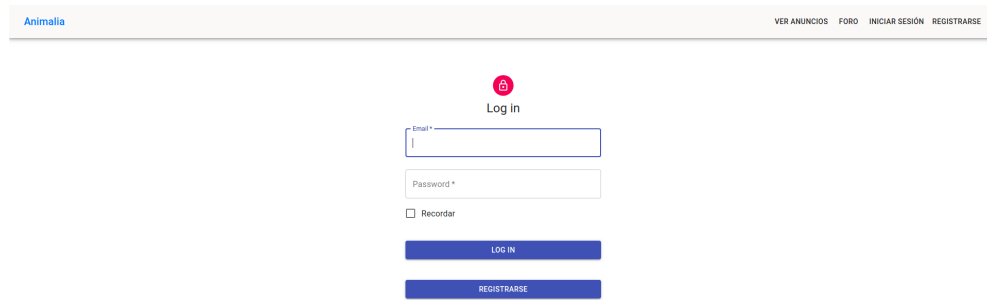
Nombre _____ Apellidos _____ Email _____ Dirección _____

CP _____ Teléfono _____ Contraseña _____ Repetir contraseña _____

CONFIRMAR REGISTRO

Figura 1: Páxina de rexistro da aplicación web

Logo de completar o proceso de rexistro procédese a loguearse na aplicación, cubrindo o email e o contrasinal.



The screenshot shows the login page of the 'Animalia' web application. At the top, there is a navigation bar with the 'Animalia' logo on the left and links for 'VER ANUNCIOS', 'FORO', 'INICIAR SESIÓN', and 'REGISTRARSE' on the right. In the center, there is a 'Log in' section with a red circular icon containing a white 'G'. Below this, there are two input fields: 'Email *' and 'Password *'. A checkbox labeled 'Recordar' is positioned below the password field. At the bottom of the login section, there are two blue buttons: 'LOG IN' and 'REGISTRARSE'.

Figura 2: Páxina de login da aplicación web

Unha vez rexistrados e logueados na aplicación web, poderemos ver a páxina principal, cos diferentes tipos de anuncios. Esta páxina correspóndese tamén coa páxina de inicio da aplicación, que se podera acceder sen rexistro. Poderemos elixir ver os anuncios de ese tipo ou crear un anuncio do tipo seleccionado.

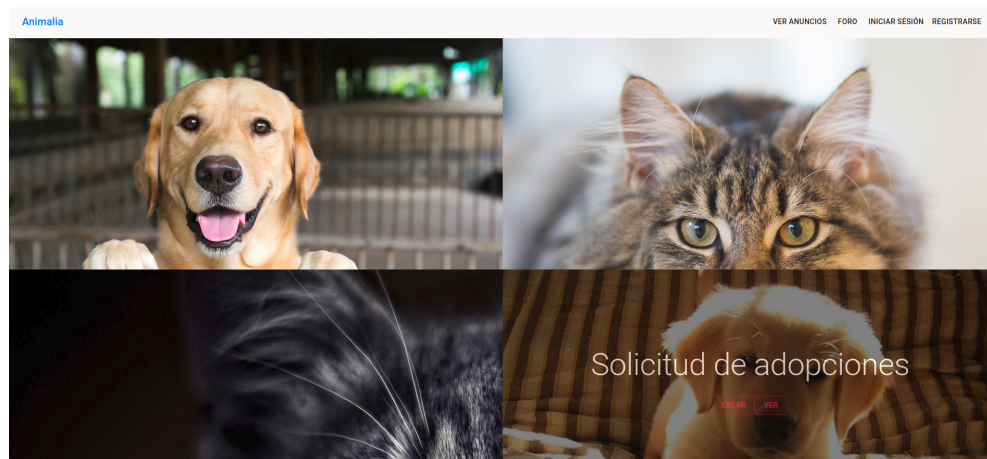


Figura 3: Páxina de inicio da aplicación web

Se seleccionamos a opción de crear anuncio, debemos cubrir os seguintes apartados sobre o animal que vamos a crear o anuncio.

Bienvenido a Animalia Juan

VER ANUNCIOS MIS ANUNCIOS MIS ANIMALES FORO SALIR


✓ Seleccionar animal

1 Registrar animal

2 Crear anuncio

3 Anuncio creado

Subir foto del animal



dog.4120.jpg

Browse

Nombre del animal *

Toby

Tipo animal

PERRO

Selecciona la raza de perro

Galgo Afgano

Sexo

MACHO

Fecha de nacimiento del animal (opcional)

10/12/2016

Descripción del animal

E un galgo de cor negra, ten un antoxo no rabo branco e moi manso.

ATRÁS

SIGUIENTE

Figura 4: Páxina de rexistro dun animal na aplicación web

Unha vez cubertos os datos sobre o animal, poderemos encher os datos sobre o anuncio que estamos cubrindo.

Bienvenido a Animalia Juan

VER ANUNCIOS MIS ANUNCIOS MIS ANIMALES FORO SALIR


✓ Seleccionar animal

✓ Registrar animal

1 Crear anuncio

2 Anuncio creado

Foto del animal



Título del anuncio *

Perdín a Toby no paseo marítimo da Coruña

Fecha del suceso

05/09/2019

Tamaño

MEDIANO

Localización

A Coruña

Descripción anuncio

Perdín a toby polo paseo da Coruña, se o atopar, avísame ap teléfono ou escribame un comentario no anuncio.

ATRÁS

SIGUIENTE

Figura 5: Páxina de rexistro dun anuncio na aplicación web

Se na páxina de inicio, Figura 3 seleccionamos a opción Ver en algún tipo de anuncio, mostraranse os anuncios en particular para ese en concreto, na seguinte figura vemos un exemplo dos animais encontrados.

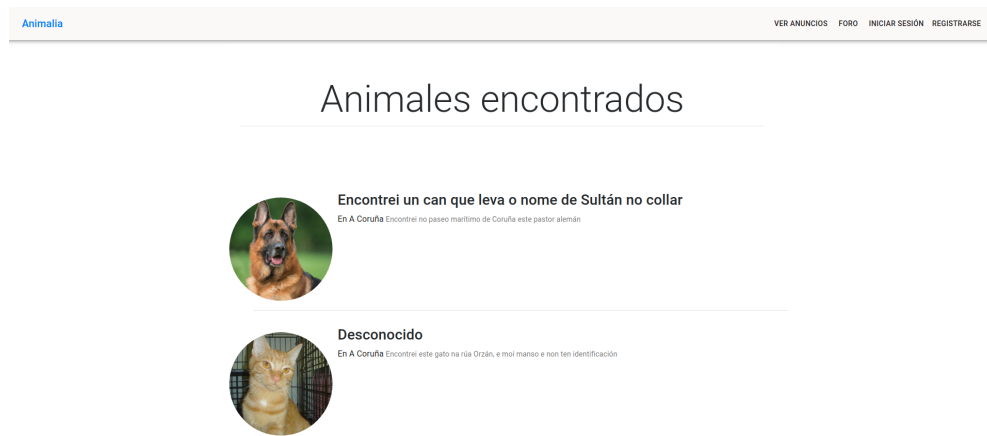


Figura 6: Páxina de animais encontrados da aplicación web

Se seleccionamos algún dos anuncios mostrados, poderemos ver a descrición dun anuncio en particular, os datos do creador do anuncio e o sistema de comentarios de cada anuncio.

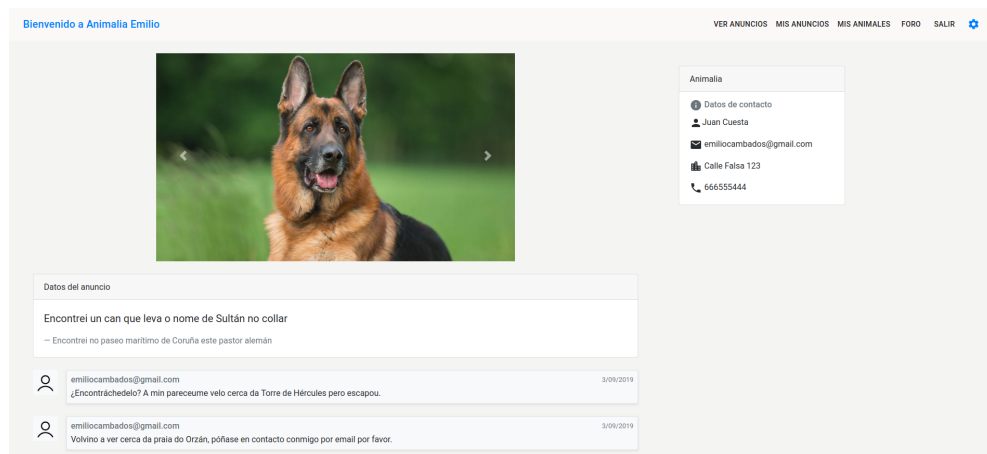


Figura 7: Páxina dun anuncio en particular da aplicación web

Por último vemos na seguinte figura unha imaxe dun fío do foro, no que un usuario lo-gueado pode comentar os fíos e así participar nunha conversa que outro usuario comezara.

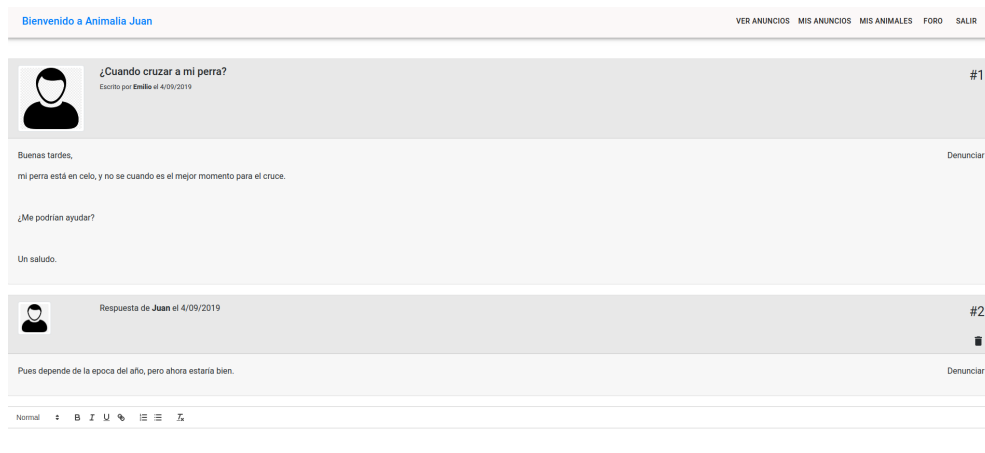


Figura 8: Páxina dun fío en particular da aplicación web

Aplicacion móbil

A continuación indicaranse un pequeno manual sobre a aplicación móbil.

Esta primeira imaxe móstrase ao iniciar a aplicación Android, nela móstranse todos os tipos de anuncios que temos na aplicación.

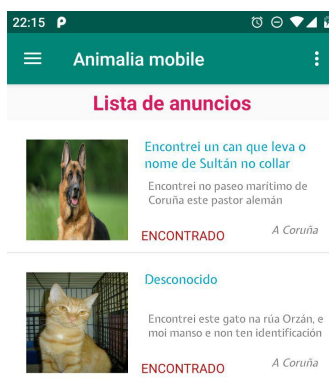


Figura 9: Páxina de inicio da aplicación aplicación móbil

Se seleccionamos da lista un anuncio podemos ver o anuncio coa foto en grande e diferentes datos do anuncio en cuestión.



Figura 10: Páxina primeria dun anuncio en particular da aplicación móvil



Figura 11: Páxina segunda dun anuncio en particular da aplicación móvil

E por último, vemos a lista de fíos disponibles e a selección dun en particular.

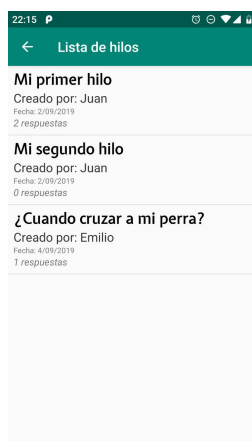


Figura 12: Páxina da lista de fío da aplicación móvil



Figura 13: Páxina dun fío en particular da aplicación móvil

Relación de Acrónimos

MVC *Model View Controller.*

API *Application Programming Interface*

POJO *Plain Old Java Object*

IDE *Integrated Development Environment*

Bibliografía

- [1] S. E. Judith Bowman and M. Darnovsky, *The Practical SQL Handbook: Using SQL Variants*, 4th ed. Addison-Wesley Professional, 2001.
- [2] B. Eckel, *Piensa en Java*, 4th ed. Grupo Anaya Publicaciones Generales, 2007.
- [3] —, *Spring*, 4th ed. Anaya Multimedia/Manning, 2015.
- [4] J. J. Villar, *Spring*, 1st ed. Kindle, 2018.
- [5] K. Simpson, *You Don't Know Js*. O'Reilly, 2014.
- [6] J. T. Gironés, *El gran libro de Android*, 7th ed. Marcombo, 2019.
- [7] I. JACOBSON, G. BOOCH, and J. RUMBAUGH, *El proceso unificado de desarrollo de software*, 2000.
- [8] R. S. Pressman, *Ingeniería del Software. Un enfoque práctico*, 2nd ed. McGraw Hill, 2010.
- [9] L. Dewailly, *Building a RESTful Web Service with Spring*, 2nd ed. Packt Publishing, 2017.
- [10] M. Fowler, *Patterns of Enterprise Application Architecture*, 1st ed. Addison-Wesley, 2002.
- [11] R. S. Pressman, *Spring Data: Modern Data Access for Enterprise Java*, 1st ed. O'Reilly, 2012.

